DOT/FAA/AR-95/7

Office of Aviation Research
Washington, D.C. 20591

# Stochastic Modeling of Antisymmetric Buffet Loads on Horizontal Stabilizers in Massively Separated Flows



March 1996

Final Report

This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

U.S. Department of Transportation
**Federal Aviation Administration**

## NOTICE

**Technical Report Documentation Page**

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| DOT/FAA/AR-95/7 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| Stochastic Modeling of Antisymmetric Buffet Loads on Horizontal Stabilizers in Massively Separated Flows | March 1996 |
| | 6. Performing Organization Code |
| 7. Author(s) | 8. Performing Organization Report No. |
| Saeed Farokhi, Clay S. Mauk, and James E. Locke | |
| 9. Performing Organization Name and Address | 10. Work Unit No. (TRAIS) |
| Aerotech Engineering & Research Corporation<br>3125 W. 6th St., Suite C<br>Lawrence, Kansas 66049-3101 | MNS 199, RPI 331 |
| | 11. Contract or Grant No.<br>DTRS-57-94-C-00059 |
| 12. Sponsoring Agency Name and Address | 13. Type of Report and Period Covered |
| U.S. Department of Transportation<br>Federal Aviation Administration<br>Office of Aviation Research<br>Washington, DC 20491 | |
| | 14. Sponsoring Agency Code<br>AAR-430 |

15. Supplementary Notes

This work was performed under the Small Business Innovation Research (SBIR) program, Phase I. The Federal Aviation Administration technical manager is Thomas DeFiore, AAR-432

16. Abstract


Federal Aviation Regulation (FAR) 25 has been revised to clearly identify buffet loading as a structural design load. A modern method was developed to model antisymmetric buffet design loads on horizontal stabilizers with a known probability, utilizing a rigid wind tunnel model. The surface pressure was measured for a large number of test conditions, including the most severe buffeting environment for this type of aircraft: the tail immersed in the massively separated wake of the wing. Modifications to the Beech King Air 200 one-sixth scale wind tunnel model included the construction of a new horizontal stabilizer instrumented with 12 miniature pressure transducers. Structural characteristics of the full-scale aircraft were estimated using the Automated STRuctural Optimization System (ASTROS) program. ASTROS, while not directly supporting buffet calculations, is written in a flexible high-level language and thus easily adaptable. Motion-dependent aerodynamics (stiffness and damping) were computed using the proven doublet lattice method, which is incorporated into ASTROS. Due to difficulties with the data acquisition system, the current approach was validated with buffet pressure power spectral densities from an existing reference. Based on the results, the methodology is sound.

Future work would include more detailed investigations of the buffet phenomenon and integration of software programs to build a multidisciplinary design tool. This would allow aircraft manufacturers to predict the horizontal stabilizer antisymmetric buffet loads early in the certification program.


| 17. Key Words | 18. Distribution Statement |
|---|---|
| Flight Loads, Structural Failure, Certification | Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161 |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 77 | |

TABLE OF CONTENTS


| | Page |
|---|---|
| EXECUTIVE SUMMARY | vii |
| 1. INTRODUCTION | 1 |
| 2. BUFFET PREDICTION METHODS | 1 |
| 2.1 Experimental Methods | 2 |
| 2.2 Data Postprocessing And Analysis | 6 |
| 2.2.1 Ideal Case | 6 |
| 2.2.2 Alternate Procedure | 7 |
| 2.3 Aeroelastic Analysis | 14 |
| 2.3.1 Modal Analysis | 15 |
| 2.3.2 Unsteady Aerodynamics | 17 |
| 2.3.3 Buffet | 18 |
| 3. RESULTS | 19 |
| 3.1 Buffet Forcing Function (Experimental) | 19 |
| 3.2 Aeroelastic Results (Analytical) | 21 |
| 4. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK | 22 |
| 5. REFERENCES | 24 |
| APPENDIXES | |
| A–TEST MATRIX AND TEST CONDITIONS | |
| B–FORTRAN PROGRAMS | |
| C–ASTROS INPUT FILES | |

LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 1 | Schematic Representation of Buffet Prediction | 2 |
| 2 | Beech King Air Model in Wind Tunnel | 3 |
| 3 | Top View of Wind Tunnel Model | 3 |
| 4 | Horizontal Stabilizer | 4 |
| 5 | Schematic Representation of Pressure Transducer Locations (X Indicates Bad Transducer, Box Indicates Lost During Testing) | 4 |
| 6 | Pressure Power Spectral Density [5] | 5 |
| 7 | Ideal Forcing Function Prediction | 6 |
| 8 | Alternate Procedure | 8 |
| 9 | Frequency Response of Low-Pass Filter | 8 |
| 10 | Characteristic Noise Floor Power Spectral Density | 9 |
| 11 | Characteristic Power Spectral Density (Run 10, Point 5, Channel 0, Unfiltered) | 10 |
| 12 | Schematic of a Linear System | 11 |
| 13 | Representative Characteristics of White Noise | 11 |
| 14 | Frequency Response Function of the Data Acquisition System (Channels 0, 1, and 2) | 11 |
| 15 | Frequency Response Functions of the Data Acquisition System (Channels 5, 6, and 8) | 12 |
| 16 | Frequency Response Functions of the Data Acquisition System (Channels 9, 10, and 11) | 12 |
| 17 | Effect of Input on Frequencies Response Functions (Channel 0) | 13 |
| 18 | Structural Drawing of Beech Super King Air 200 [8] | 16 |

19 Finite Element Model of the Beech King Air 200 Horizontal Stabilizer 17

20 Horizontal Stabilizer Structural Mode Shapes 18

21 Buffet Pressure Power Spectral Densities $M = 0.8$, $\Lambda = 25°$ [10] 20

22 Computed Buffet Pressure PSD (Run 10, Point 5, Channel 0) 20

23 Computed Bending Moment for the Beech King Air 200 (Run 10, Point 5, Left Side) 21

24 Computed Bending Moment for the Beech King Air 200 with Buffet Pressure PSDs Equal $10^{-4}$ 21

25 Predicted Bending Moment PSDs [3] F-18 Vertical Tail 12 Percent Wind Tunnel Model V=81.5 ft/sec 22

# EXECUTIVE SUMMARY

Federal Aviation Regulation (FAR) 25 has been revised to clearly identify buffet loading as a structural design load. A modern method was developed to model antisymmetric buffet design loads on horizontal stabilizers with a known probability, utilizing a rigid wind tunnel model. The surface pressure was measured for a large number of test conditions, including the most severe buffeting environment for this type of aircraft: the tail immersed in the massively separated wake of the wing. Modifications to the Beech King Air 200 one-sixth scale wind tunnel model included the construction of a new horizontal stabilizer instrumented with 12 miniature pressure transducers. Structural characteristics of the full-scale aircraft were estimated using the ASTROS (Automated STRuctural Optimization System) program. ASTROS, while not directly supporting buffet calculations, is written in a flexible high-level language and thus easily adaptable. Motion-dependent aerodynamics (stiffness and damping) were computed using the proven doublet lattice method, which is incorporated into ASTROS. Due to difficulties with the data acquisition system, the current approach was validated with buffet pressure power spectral densities from an existing reference. Based on the results, the methodology is sound.

This work was performed under the Small Business Innovation Research (SBIR) Program, Phase I. Future work would include more detailed investigations of the buffet phenomenon and integration of software programs to build a multidisciplinary design tool. This would allow aircraft manufacturers to predict stabilizer antisymmetric buffet loads early in the certification program.

## 1. INTRODUCTION.

The Federal Aviation Administration (FAA) has a continuing program to collect data and develop predictive methods for aircraft flight loads. Some of the most severe and potentially catastrophic flight loads are produced by separated flows. Structural response to the aerodynamic excitation produced by separated flows is defined as buffeting [1]. Buffeting can cause serious controllability problems and in severe cases produce structural failure. The result of control difficulties can be catastrophic if the aircraft is in a near ground flight path such as landing or takeoff. Structural failure, in the extreme, is life-threatening at any flight condition.

The potential severity of tail buffet has persuaded the FAA to include buffet loading as a design load criterion for commercial transports. Under Federal Aviation Regulation (FAR) 25-305(e), aircraft manufacturers are required to demonstrate that the cumulative probability of an aircraft encountering dangerous levels of buffet-induced rolling moment is below the prescribed level. The current accepted method of meeting this requirement involves a great deal of full-scale flight testing. This method costs manufacturers a large amount of capital to meet the requirement and allows them no easy recourse should the aircraft not qualify. New methodologies are being considered that would allow the design rolling moment load to be estimated before the full-scale aircraft is constructed. A standardized method would expedite the certification process and enable consistent and repeatable results.

Two major classes of buffet prediction methods are currently in use. The first method is buffet prediction by computational fluid dynamics codes. This method is very computationally intensive, requires an expert user, and is still unproven. An alternate approach is to use experimental data in conjunction with a computational solution of the structural dynamics equations. Experimental/computational methods also have several subdivisions, most notably in the experimental methods employed. The wind tunnel model used for measurement of the unsteady surface pressure can be rigid or flexible. The merits of each type of buffet prediction methodology are summarized in references 2, 3, and 4.

This study describes an experimental/computational method to model the antisymmetric buffeting of horizontal stabilizers in massively separated flows. The objective is to predict, within a known probability, the antisymmetric response. The most obvious benefit is safety. If an aircraft has predictable characteristics in critical flight scenarios, precautionary measures can be taken. If a prediction of undesirable behavior can be performed early in the design process, it can be remedied.

## 2. BUFFET PREDICTION METHODS.

A rigid body method for buffet prediction was chosen due to its relatively low cost and experimental simplicity. The prediction methodology can be divided into two distinct tasks: (1) experimental acquisition of unsteady pressure on the tail of a rigid model, and (2) prediction of the aeroelastic results based on the buffet forcing function as defined by the first task. The prediction of tail buffet using this methodology can be best summarized in the flowchart illustrated in figure 1. Each segment of the flowchart will be discussed in the sections to follow.

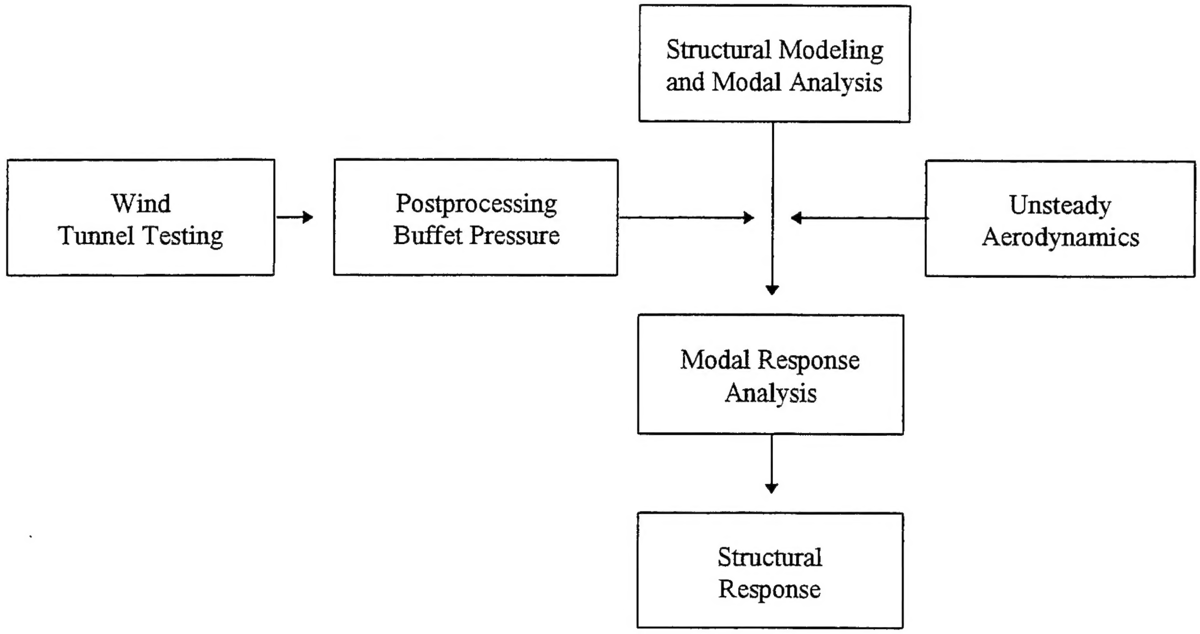


FIGURE 1. SCHEMATIC REPRESENTATION OF BUFFET PREDICTION

## 2.1 EXPERIMENTAL METHODS.

The experimental testing, under arrangement and independent contract by the FAA, was performed at National Institute of Aviation Research (NIAR) facilities in Wichita, Kansas. As prescribed by the FAA, the model used for this phase of the work was a one-sixth scale Beech King Air 200. The King Air 200 is a t-tail configuration as shown in figure 2. The model used for this experiment had variable elevator, rudder, and flap deflections. It was tested in the Walter H. Beech Memorial Tunnel which has a 7- by 10-ft test section. Due to the large scale of the model, the model was tested with the wing tips removed to allow some wall clearance (approximately 8 in.) and avoid the wall boundary layer, as shown in figure 3. The new horizontal stabilizer was manufactured by the NIAR engineering machine shop and instrumented as illustrated in figure 4. The pressure transducers used were Kulite LQ-125-5SG with the following characteristics: measuring range 0-5 psi, nonlinearity and hysteresis ± 0.5 percent, and natural frequency 70-350 KHz. The placement of the pressure transducers was symmetrical and located as schematically described in figure 5. Although passing pretests, transducers 3, 4, and 7 failed to perform dynamically. Transducer 9 failed similarly during the test.

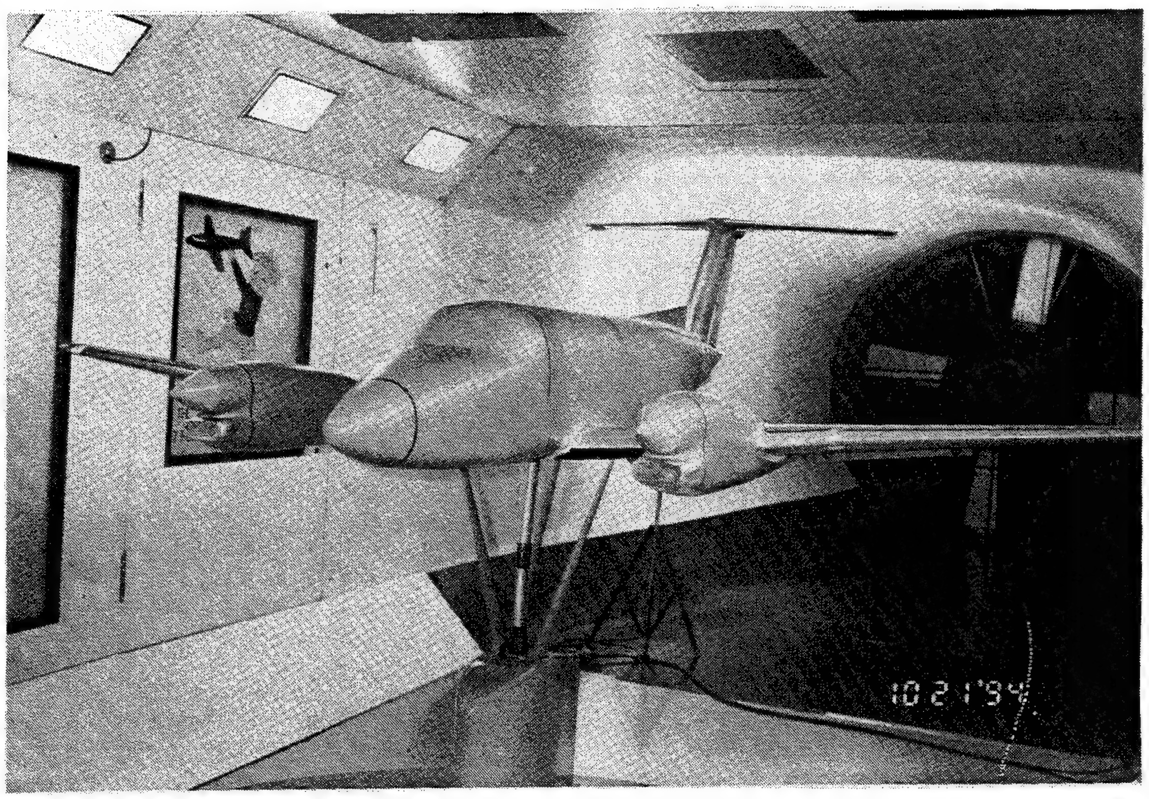

FIGURE 2. BEECH KING AIR MODEL IN WIND TUNNEL

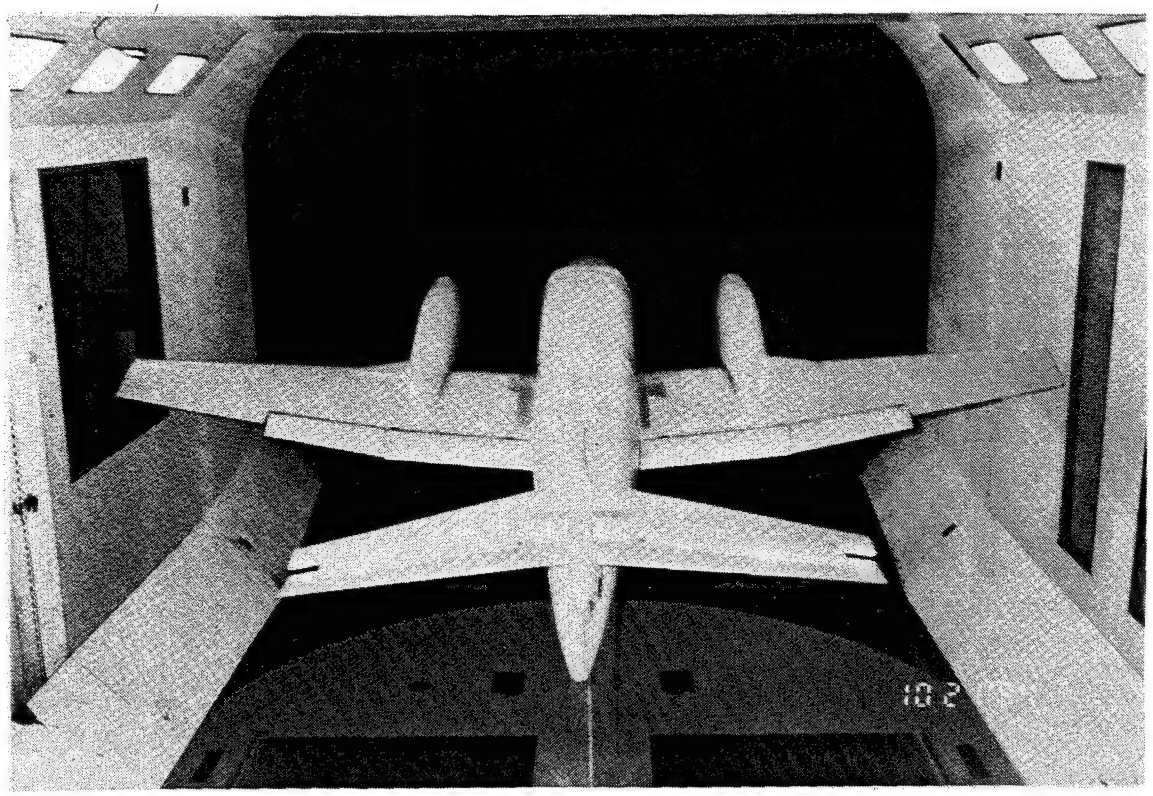

FIGURE 3. TOP VIEW OF WIND TUNNEL MODEL

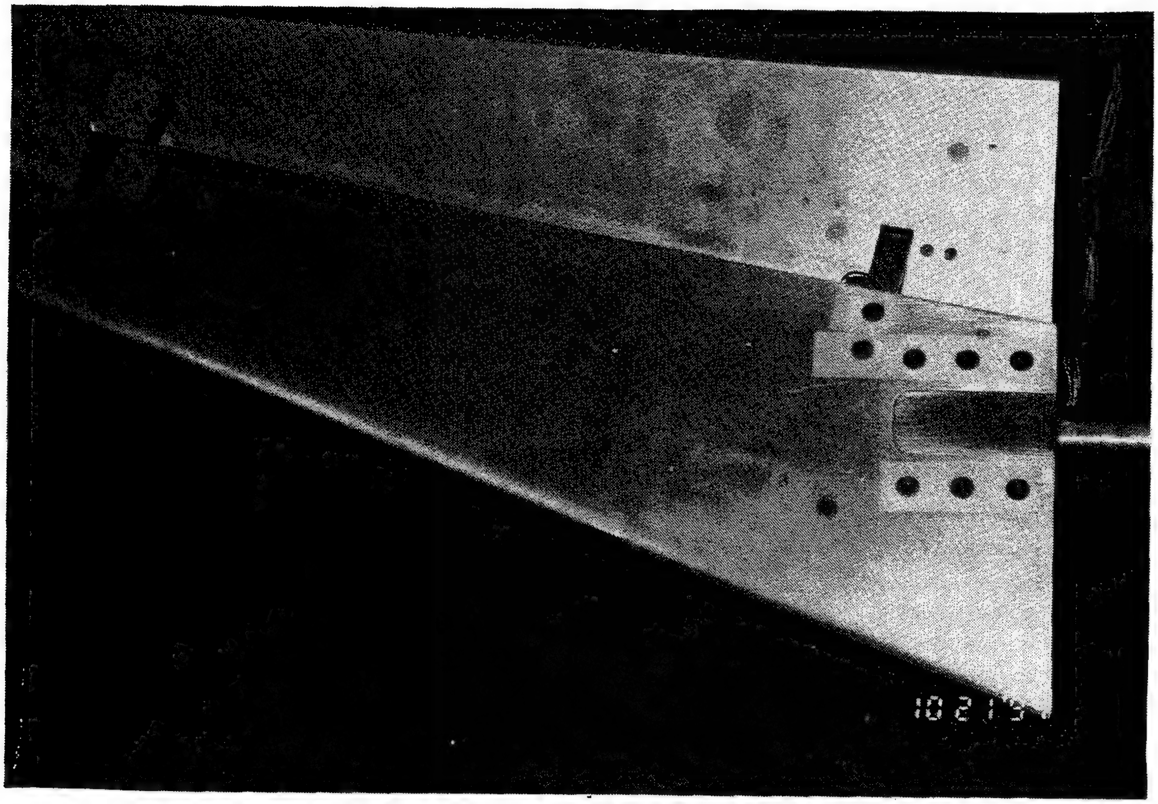

FIGURE 4. HORIZONTAL STABILIZER

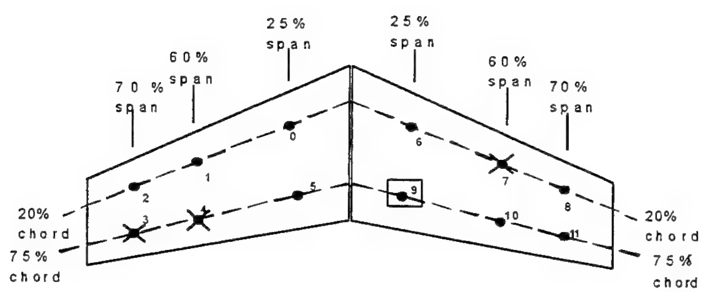


FIGURE 5. SCHEMATIC REPRESENTATION OF PRESSURE TRANSDUCER LOCATIONS (X INDICATES BAD TRANSDUCER, BOX INDICATES LOST DURING TESTING)

The selection of the data acquisition parameters is crucial to insure that an adequate quantity of data is collected in the proper frequency range. Based on published experimental results of

buffet pressures that show peaks near 1000 Hz (see figure 6) for near transonic wing buffet, the sampling frequency was originally prescribed at 4000 Hz (with a low-pass filter, cutoff 2000 Hz). A sampling frequency of 4000 Hz translates into data up to the Nyquist frequency, 2000 Hz. In a random experiment, the number of samples becomes crucial in that if more cycles are averaged, the random error is lowered (according to reference 6, random error of power spectral density is given by $\varepsilon = \sqrt{\frac{1}{n_d}}$ where $n_d$ is the number of cycles). Due to the inability of NIAR's data acquisition system to perform to the advertised level, and their decision not to perform any data reduction, the parameters were adjusted. The data acquisition parameters, as tested, were a sampling frequency of 1210.54 Hz, for 8192 samples. This defines the Nyquist frequency at approximately 605 Hz. Ideally a low-pass filter should have been applied with a cutoff frequency near 600 Hz to prevent aliasing.

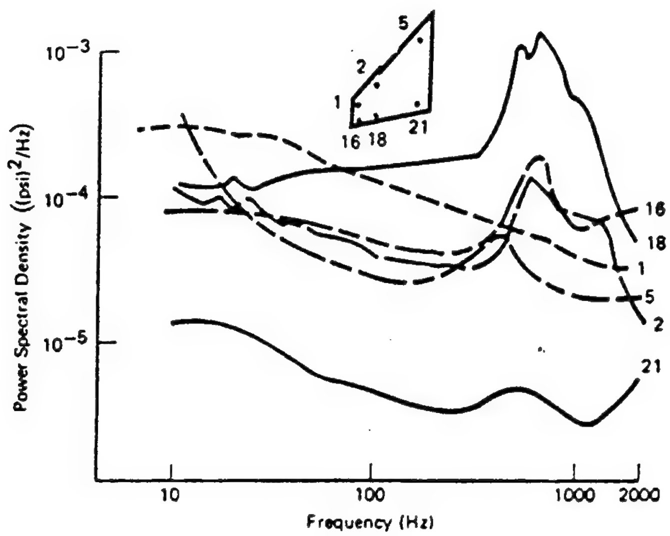


FIGURE 6. PRESSURE POWER SPECTRAL DENSITY [5]

The test matrix was proposed as:

- 3 dynamic pressures (25, 30, and 41 psf)
- 2 flap settings (up, full down {35°})
- 3 rudder settings (0°, 10°, and 20°)
- 3 elevator settings (-20°, 0°, and 20°)
- 3 heading angles (0°, -10°, and -20°)
- 6 angles of attack (0°, 5°, 10°, 15°, 20°, and 25°)

The actual test matrix varied slightly, primarily in the number of test runs completed: the principal cause being time limitations in conjunction with the model pushing the limits of the tunnel balance system. The matrix of conditions tested is included in appendix A, which also

contains the computed physical test characteristics of the test runs, including velocity, Mach number, and Reynold's number.

To establish baseline performance of the data acquisition system, two experimental runs were made. Both of these runs were conducted at zero tunnel velocity and were the first of the day to avoid any thermal effects. These experimental runs (1 and 70) will hereafter be referred to as noise floor runs. Data acquired during these runs should ideally be zero.

## 2.2 DATA POSTPROCESSING AND ANALYSIS.

The goal of postprocessing was to convert the time series data measured by NIAR into a useful format for the remaining computational tail buffet estimation. Difficulties were encountered during the postprocessing due to the shortcomings of the data acquisition system. In particular, the data acquisition system transfer function was nonlinear and demanded a special alternate approach.

### 2.2.1 Ideal Case.

As schematically illustrated in figure 7, the ideal case for estimating the power spectral densities of time series is a methodical and logical procedure. Postprocessing begins immediately after the recording of the pressure data (dashed outline). It should be noted that the low-pass filter prevents any biasing in the data by higher frequencies (above the Nyquist frequency).

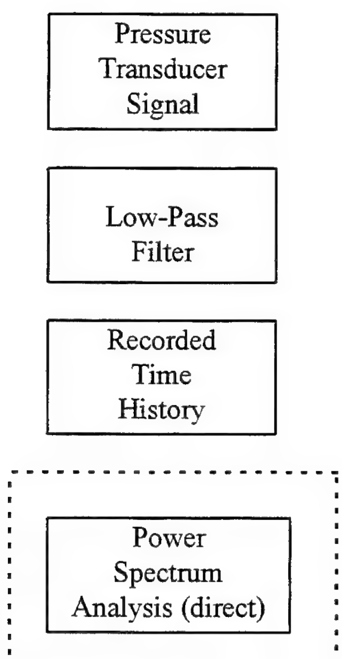


FIGURE 7. IDEAL FORCING FUNCTION PREDICTION

The most computationally efficient and popular method for estimating the power spectral densities of time data is the Fast Fourier Transform. This also allows for simple implementation of an intermediate stage where the noise floor can be subtracted in the frequency domain. For uncorrelated noise, this can be represented mathematically as Y(f) = X(f) + N(f), where Y(f) is the measured signal, X(f) is the desired signal, and N(f) is the uncorrelated noise. The data is divided into $n_d$ segments and the one-sided power spectral density is computed as

$$G_{xx}(f) = \frac{2}{n_d T} \sum_{i=1}^{N} \left| X_i(f,T) \right|^2 \tag{1}$$

where T = Sample Period
$X_i(f,T)$ = Fourier transform of $x_i(t,T)$

The cross correlation is an equally simple equation:

$$G_{xy}(f) = \frac{2}{n_d T} \sum_{i=1}^{N} X_i^*(f,T) Y_i(f,T) \tag{2}$$

* denotes complex conjugate

where $Y_i(f,T)$ = Fourier transform of $y_i(t,T)$

2.2.2 Alternate Procedure.

Due to polluted data received from the testing at NIAR, an alternate procedure was used to attempt a recovery of some usable data from the time histories. Figure 8 illustrates the procedure used by Aerotech during this Phase I investigation (dashed outline). Due to NIAR's inability to provide an analog low-pass filter to prevent aliasing, the first step is to apply a digital low-pass filter in the time domain. In accordance with the overall goal of analyzing the frequency difference between different locations, any filter applied must be a linear phase filter. Aerotech applied a finite impulse response (FIR) filter for this purpose. The Fortran 77 code used to compute the coefficients of the low-pass FIR filter is included in appendix B. This type of filter ensures that any changes to the phase of the data will be at most linearly shifted and not altered in any unpredictable manner. Figure 9 is the frequency response of the low-pass filter applied to the data.

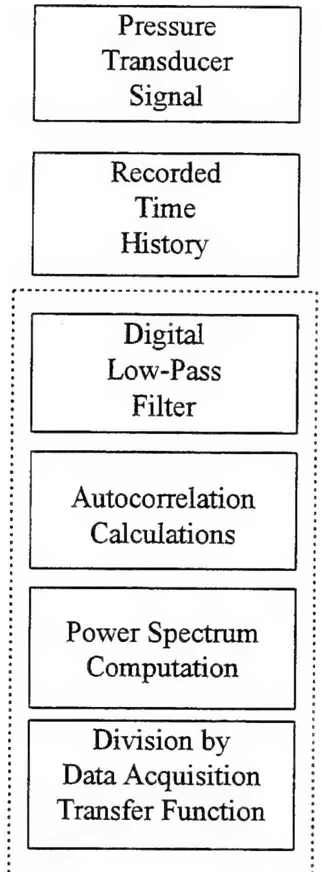


FIGURE 8. ALTERNATE PROCEDURE

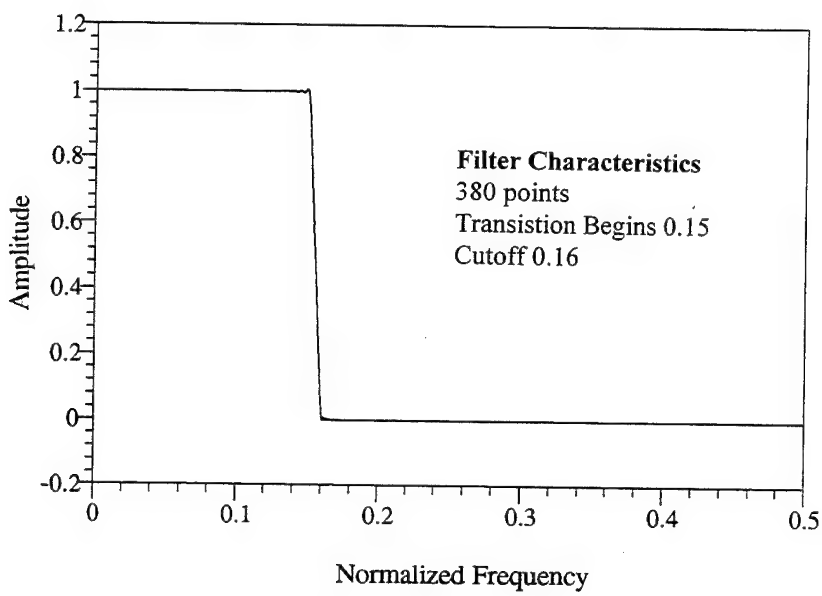


FIGURE 9. FREQUENCY RESPONSE OF LOW-PASS FILTER

The second step in the postprocessing procedure shown in figure 8 requires some background information. As mentioned in the description of the ideal case, the noise floor should be subtracted in the frequency domain to remove the uncorrelated noise. The power spectral densities of the experimental noise floor data are typified by figure 10. This figure illustrates the major difficulty in the data postprocessing of this project. The power spectral density of the noise floor is expected to be at best a relatively uniform distribution of white noise incurred due to background noise. The figure illustrates that the overall level is quite high and the data contains many frequency spikes, most notably near 200 Hz. Initial results calculated using equation 1 for transducer 0 show (figure 11) the same characteristic spikes and nearly identical response to the noise floor.

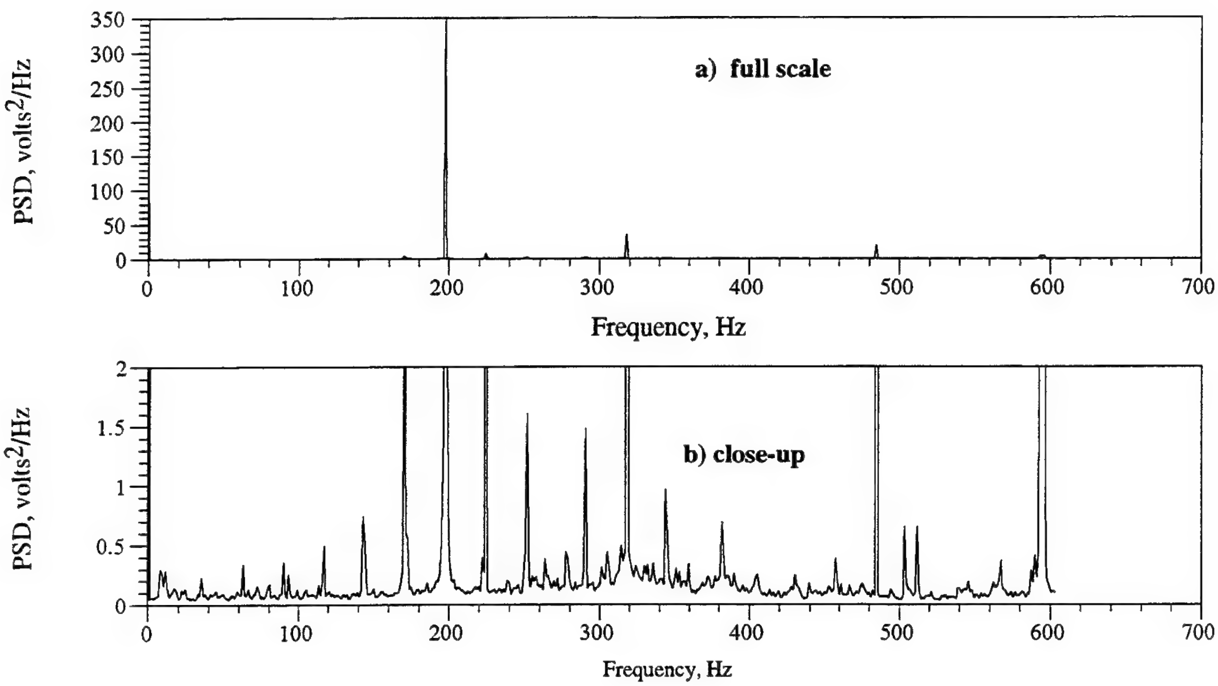


FIGURE 10. CHARACTERISTIC NOISE FLOOR POWER SPECTRAL DENSITY

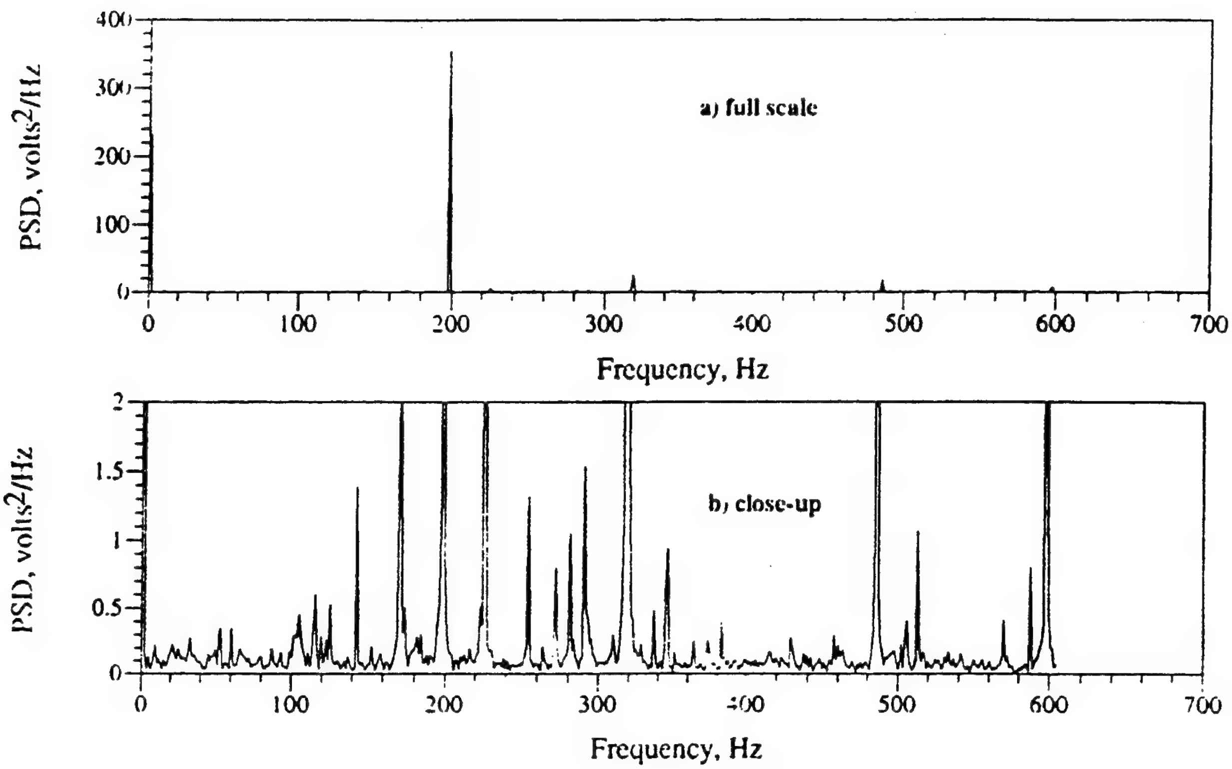


FIGURE 11. CHARACTERISTIC POWER SPECTRAL DENSITY (RUN 10, POINT 5, CHANNEL 0, UNFILTERED)

After attempts to subtract the noise floor failed, it was decided to investigate the dynamics of the data acquisition system. With the assumption that the data acquisition system could, in fact, have its own transfer function, as illustrated in figure 12, more testing was prescribed. Ideally, a data acquisition system should have a transfer function of unity. To measure the data acquisition system transfer function, a white noise generator was developed and more testing was performed by supplying white noise input to the data acquisition system. The result should have a uniform power spectral density, as shown in figure 13, and subsequently a transfer function equal to unity. The actual results from the experiments are shown in figures 14-16. Figures 14-16 show the transfer function for each channel of the data acquisition system with the input white noise having a standard deviation of 0.005. The transfer functions are not equal to unity, and each channel has a unique transfer function. Several frequency response functions of the data acquisition system (Channels 0, 1, and 2) channels were tested using different parameters. Figure 17 shows channel 0 with input parameters as noted on the figure; it should be noted that the transfer function of the data acquisition system is not consistent for a given channel with different values of input, that is, the transfer function is nonlinear. Due to time constraints, the transfer functions computed and shown in figures 14-16 were used to compute the pressure on the tail surface.

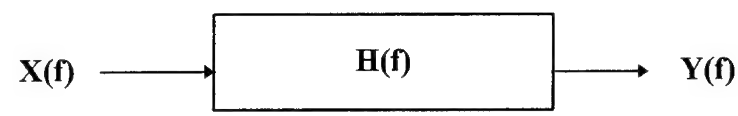


FIGURE 12. SCHEMATIC OF A LINEAR SYSTEM

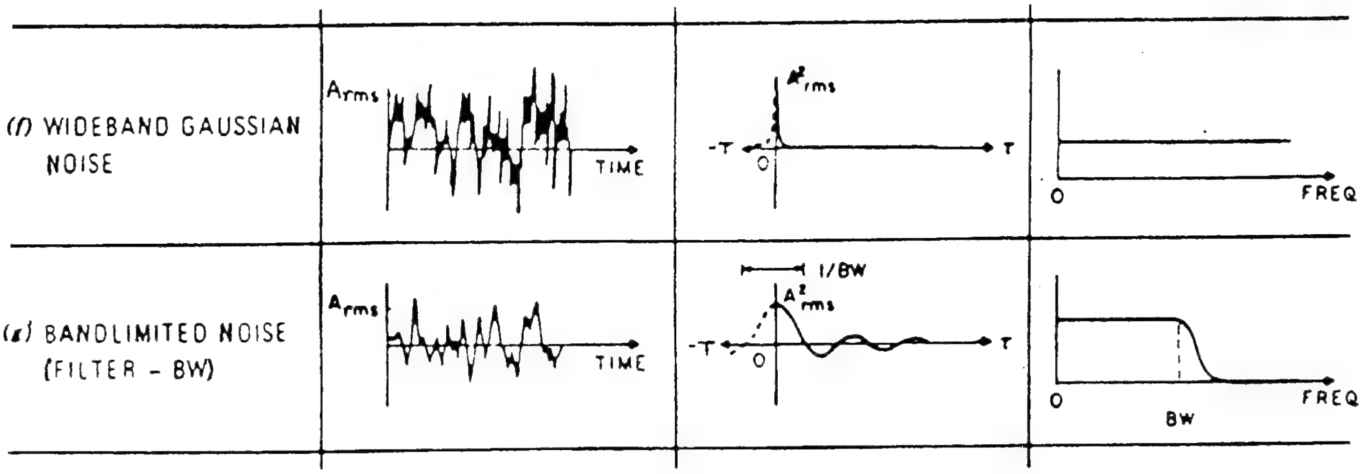


FIGURE 13. REPRESENTATIVE CHARACTERISTICS OF WHITE NOISE

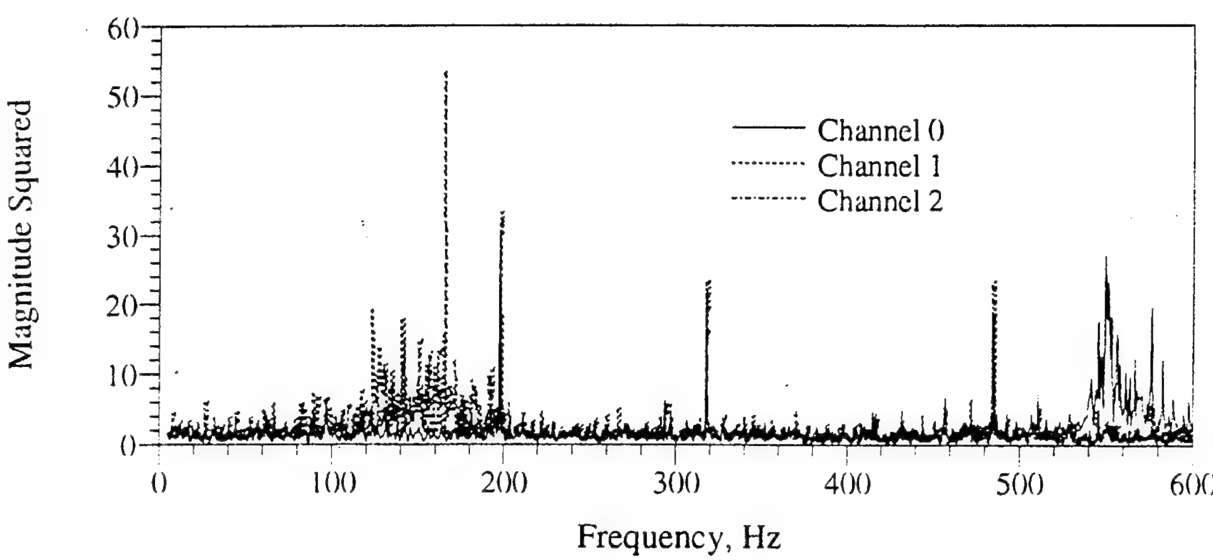


FIGURE 14. FREQUENCY RESPONSE FUNCTION OF THE DATA ACQUISITION SYSTEM (CHANNELS 0, 1, AND 2)

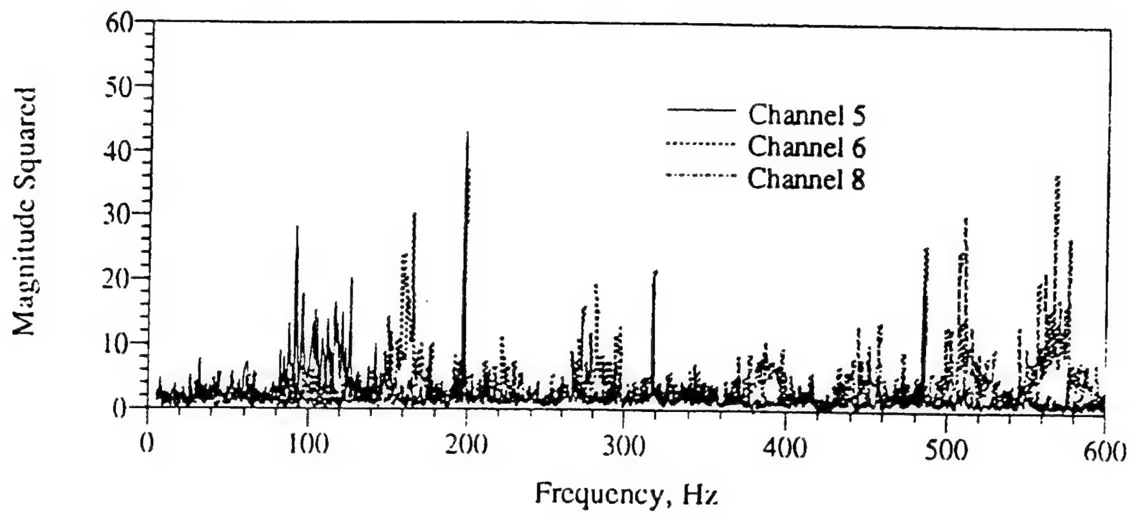


FIGURE 15. FREQUENCY RESPONSE FUNCTIONS OF THE DATA ACQUISITION SYSTEM (CHANNELS 5, 6, AND 8)

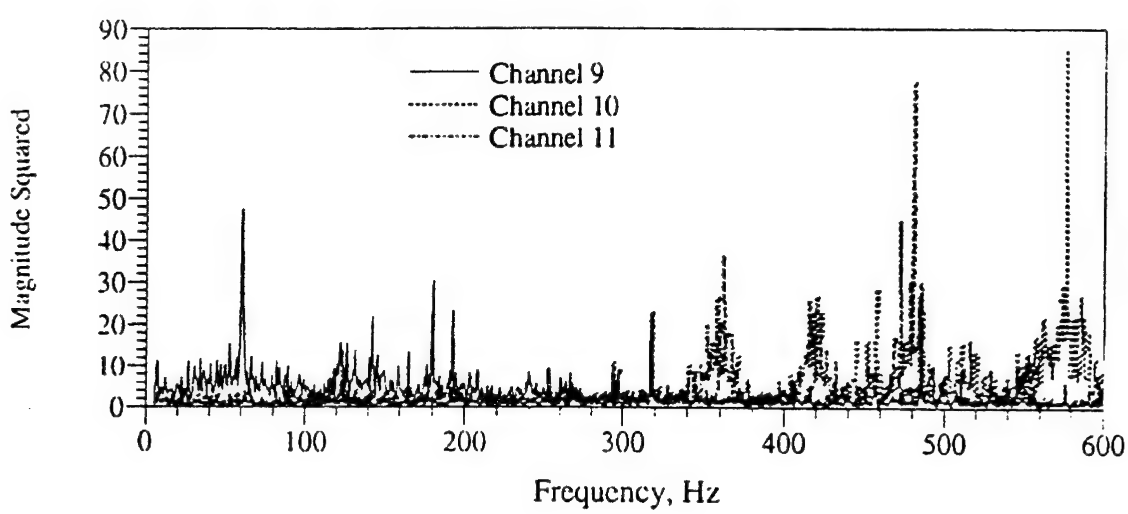


FIGURE 16. FREQUENCY RESPONSE FUNCTIONS OF THE DATA ACQUISITION SYSTEM (CHANNELS 9, 10, AND 11)

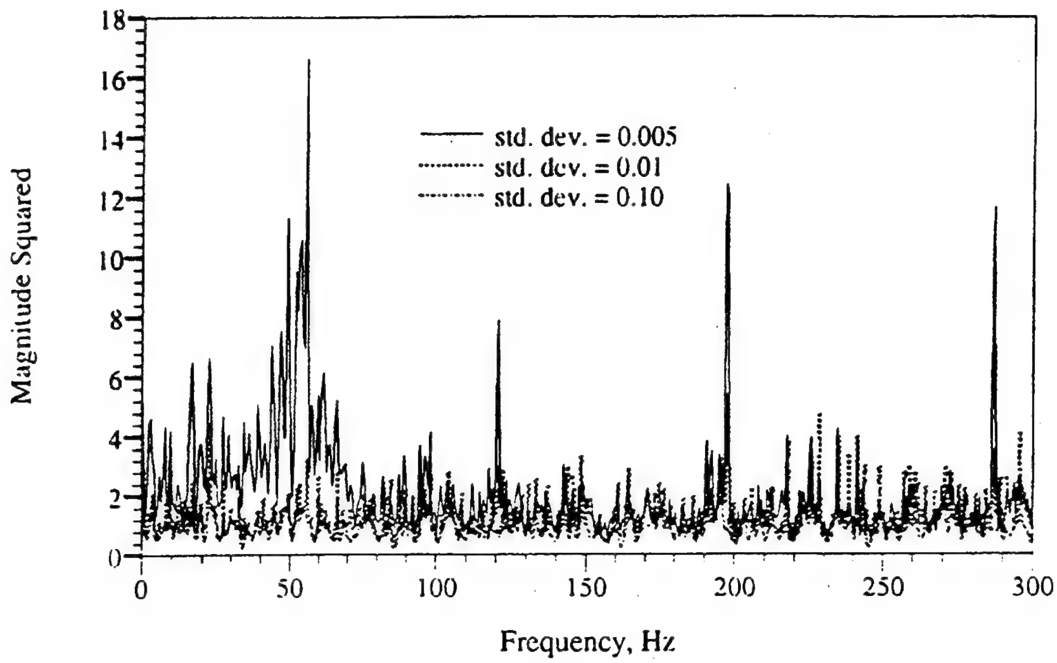


FIGURE 17. EFFECT OF INPUT ON FREQUENCIES RESPONSE FUNCTIONS (CHANNEL 0)

Another approach to compensate for the polluted data was to compute the power spectral densities indirectly using autocorrelations. This requires considerably more computational time than direct computation. The autocorrelation is defined as

$$\hat{R}_{xx}(r\Delta t) = \frac{1}{N-r}\sum_{n=1}^{N-r} x_n x_{n+r} \quad r = 0,1,2,\ldots m \tag{3}$$

where r is the lag number
m is the maximum lag number (m<N)
N is the number of samples

The cross-autocorrelation is similarly computed by

$$\hat{R}_{xy}(r\Delta t) = \frac{1}{N-r}\sum_{n=1}^{N-r} x_n y_{n+r} \quad r = 0,1,2,\ldots m \tag{4}$$

The power spectrums are computed by the autocorrelations which have been averaged over a number of segments by

$$G_{xx} = 2\,S_{xx} = \int_{-infinity}^{infinity} R_{xx}(\tau)\, e^{j2\pi f\tau}\, d\tau = Fourier\ Transform\ of\ R_{xx} \tag{5}$$

Similarly, the cross-power spectrum can be computed by

$$G_{xy} = 2\,S_{xy} = \int_{-infinity}^{infinity} R_{xy}(\tau)\,e^{j2\pi f\tau}\,d\tau = \textit{Fourier Transform of } R_{xy} \tag{6}$$

The transfer functions of the data acquisition system, as discussed above, can be accounted for using the following relations

$$\overline{G}_{xx} = \frac{G_{xx}}{H^2_{xx}} \tag{7}$$

$$\overline{G}_{xy} = \frac{G_{xy}}{H^*_{xx} H_{yy}}$$

The overbarred values represent the estimated power spectral densities of the buffet forcing function. Fortran 77 programs written to perform the above computations are included in appendix B.

## 2.3 AEROELASTIC ANALYSIS.

Buffeting is governed by the dynamic equilibrium equation 8 in terms of the generalized coordinate q:

$$M_n\ddot{q}_n(t) + D_n\dot{q}_n(t) + \omega_n^2 M_n q_n(t) + F_{D_n}\big(\dot{q}_1(t)\ldots\dot{q}_N(t)\big) + F_{K_n}\big(q_1(t)\ldots q_N(t)\big) = P_n(t) \tag{8}$$
$$n = 1\ldots N$$

where the terms are defined as

| | |
|---|---|
| $M_n$ | generalized mass |
| $D_n$ | structural damping |
| $\omega_n$ | resonance frequency |
| $\omega_n^2 M_n$ | structural stiffness |
| $F_{Dn}$ | motion-dependent aerodynamic damping |
| $F_{Kn}$ | motion-dependent aerodynamic stiffness |
| $P_n$ | motion-independent aerodynamic force (buffet pressure excitation) |

Linear modal analysis determines the terms $M_n$, $D_n$, and $\omega_n$. The aerodynamic terms can be determined using two approaches: (1) extract the terms from wind tunnel testing, or (2) compute the terms using an aerodynamics model. For this study the motion-dependent aerodynamic stiffness and damping terms were determined using the doublet lattice method. Wind tunnel testing was used to determine the buffet pressure excitation $P_n$. As discussed in earlier sections, the wind tunnel approach was chosen due to the lack of proven computational methods for buffet pressure excitation.

The ASTROS (Automated STRuctural Optimization System) program [7] is a multidisciplinary finite element-based procedure for the design and analysis of aerospace structures. It is a public domain program with proven capabilities paralleling those of NASTRAN. Analysis options include structural response (static and dynamic) and aeroelastic analysis (static and dynamic). In addition, ASTROS is written in a flexible high-level language, MAPOL (Matrix Analysis Problem Oriented Language). Although the program does not directly support buffet analysis, ASTROS can be used for buffet analysis by modifying the standard MAPOL sequence. All structural and aeroelastic terms appearing in the left-hand side of equation 1 were computed using ASTROS. The implementation is discussed in the following sections.

2.3.1 Modal Analysis.

The ASTROS modal analysis solution sequence permits the determination of structural mode shapes and frequencies. A finite element model of the Beech Super King Air 200 horizontal tail was constructed based on figure 18 and the following assumed characteristics [8, 9]:

- half span: 110 in.
- root chord: 61 in.
- tip chord: 30 in.
- root depth: 5 in.
- tip depth: 3 in.
- front spar located at 15 percent chord
- rear spar located at the elevator hinge line
- seven evenly spaced ribs
- rib web thickness: 0.040 in.
- spar web thickness: 0.040 in.
- skin thickness: 0.050 in. (includes skin stiffeners and spar caps)

The resulting finite element model is shown in figure 19, with the corresponding ASTROS input/output in appendix C. Each structural node point is represented as a GRID. Rib and spar webs were modeled as shear panels (CSHEAR elements), and the skins were modeled as membranes (constrained CQUAD4 elements). Vertical stiffeners (CROD elements) were also included to provide membrane stiffness at shear panel boundaries. The mass of the structure was included using concentrated masses (CONM2 elements) at each node.

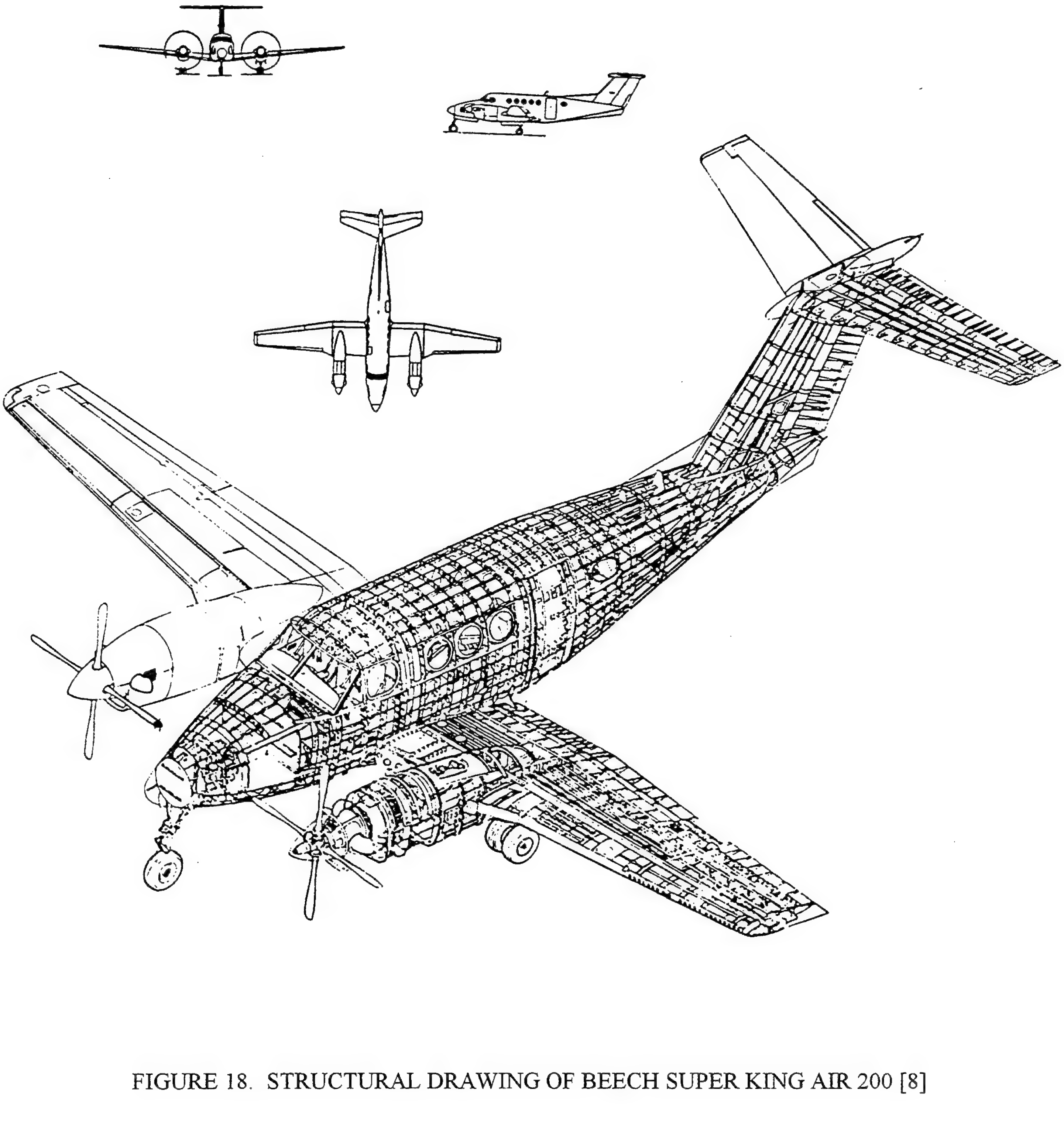

FIGURE 18. STRUCTURAL DRAWING OF BEECH SUPER KING AIR 200 [8]

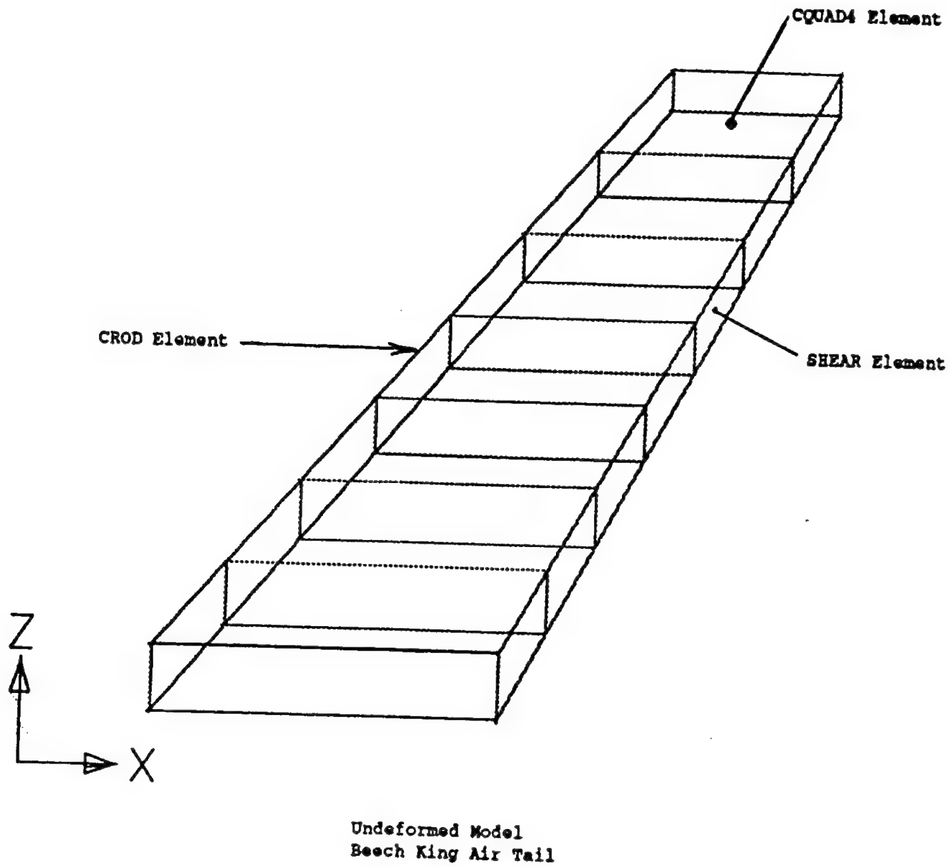


FIGURE 19. FINITE ELEMENT MODEL OF THE BEECH KING AIR 200 HORIZONTAL STABILIZER

The first three computed mode shapes and frequencies are shown in figure 20. As noted in the figure, the modes occur at 20.5 Hz (pure bending), 92.5 Hz (bending and torsional), and 121.9 Hz (torsional). Due to the low-pass filter cutoff discussed previously, only these first three modes where considered.

2.3.2 Unsteady Aerodynamics.

The ASTROS flutter, gust, and blast analyses solution sequences include the aerodynamic stiffness and damping terms of equation 8. These terms are computed by use of the doublet lattice method. As noted in reference 7 (ASTROS Applications Manual), the doublet lattice method is recognized as a standard in the aerospace industry. Based on the dimensions given in the previous section, an ASTROS aerodynamic model was created using an aerodynamic panel (CAERO1 element).

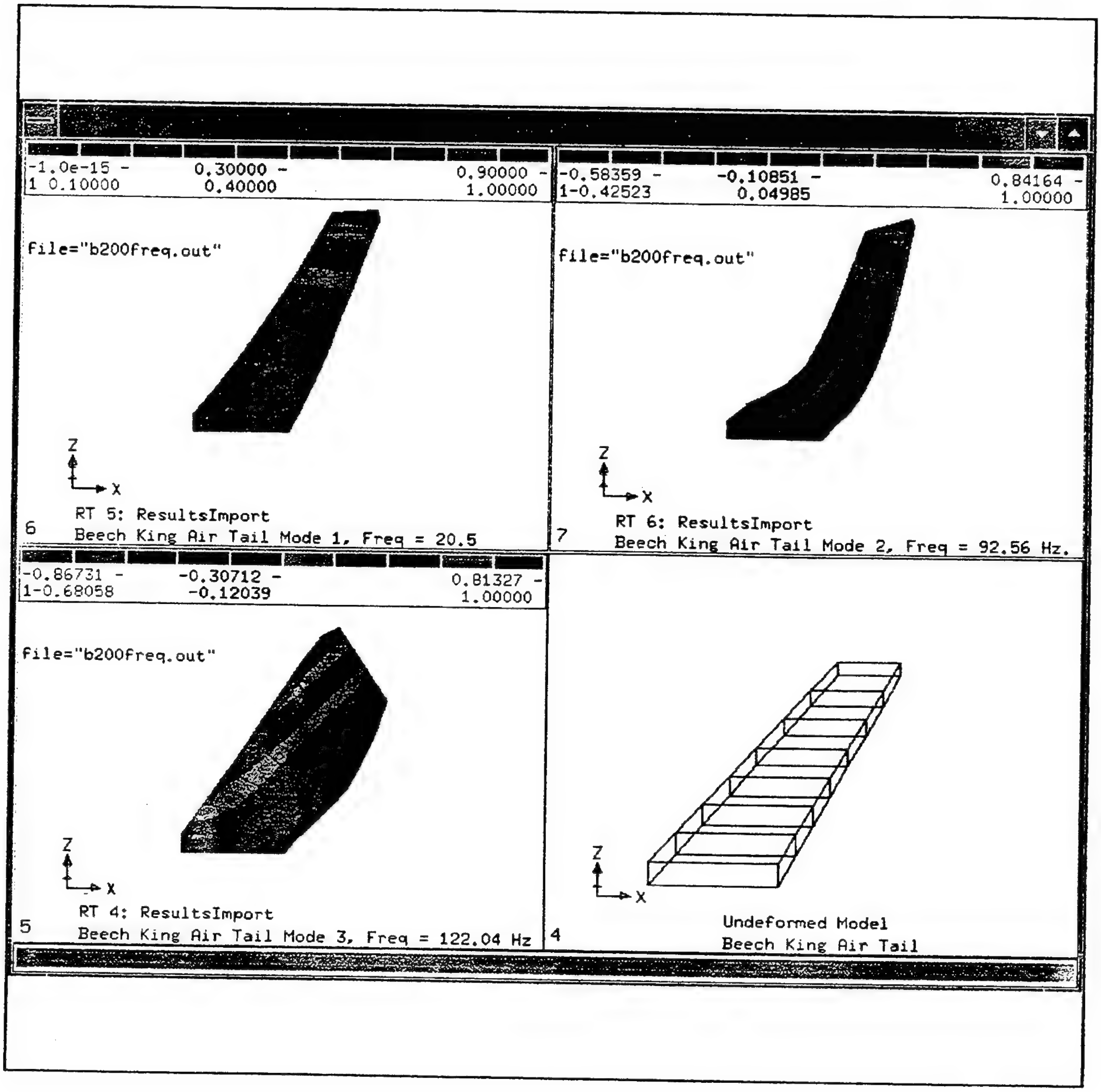


FIGURE 20. HORIZONTAL STABILIZER STRUCTURAL MODE SHAPES

2.3.3 Buffet.

There are two differences between the present problem and the ASTROS gust analysis solution sequence: (1) the right-hand side of equation 8 is different, and (2) gust analysis in ASTROS is treated as a frequency response analysis, not a random response analysis. The present buffet problem was solved using a multistep approach. Key steps in the computational process are as follows:

- Determine the modal complex frequency response matrix, $[H(\omega)]$. This matrix can be computed at each frequency of interest by replacing the gust analysis right-hand side with the identity matrix. The resulting response matrix is $[H(\omega)]$. The right-hand side identity matrix was input using the ASTROS direct matrix input (DMI).

- Use the mode shapes and the wind tunnel measured buffet pressure PSD matrix to determine the modal buffet pressure PSD matrix $[S_x(\omega)]$.

- Compute the modal response PSD matrix $[S(\omega)] = [H(-\omega)][S_x(\omega)][H(\omega)]^T$.

- Compute the desired structural response PSD $[S_y(\omega)] = [N_y]\ [S(\omega)]\ [N_y]^T$. The terms in the row matrix $[N_y]$ are the structural responses due to unit modal displacements. For example, to compute the tip displacement PSD, $N_{y_{11}}$ is the tip displacement due to $q_1 = 1$ and $q_2 = q_3 = 0$, $N_{y_{12}}$ is the tip displacement due to $q_2 = 1$ and $q_1 = q_3 = 0$, etc. For this study three PSDs were calculated: tip displacement, tip rotation, and root bending moment.

The modal complex frequency response matrix $[H(\omega)]$ and the $[N_y]$ matrices were determined using a modified gust analysis MAPOL sequence. All other computations were performed outside of ASTROS (Fortran 77 code included in appendix B). ASTROS input files are included in appendix C.

## 3. RESULTS.

### 3.1 BUFFET FORCING FUNCTION (EXPERIMENTAL).

Due to the complications and shortcomings of the data acquisition system, it was pointless to attempt parameter studies. The resulting PSDs computed using the alternate approach, as described in section 2.2.2, will only be presented in a representative manner.

For comparison, figures 6 (in section 2.1) and 21 represent buffet pressure power spectral densities from similar experimentation. As most of the previous research primarily involved wing buffet due to local separations (or vortex burst related), unsteady pressures shown in these figures are primarily due to local (wing) separations. The major aspects that can be compared from these figures are the magnitudes and the decays.

Comparing figures 6 and 21 to the results of this study, represented by figure 22, the magnitude becomes the primary obstacle. The PSDs estimated in this project are nearly four orders-of-magnitude higher. In addition, the spikes due to the data acquisition system dominate the PSD.

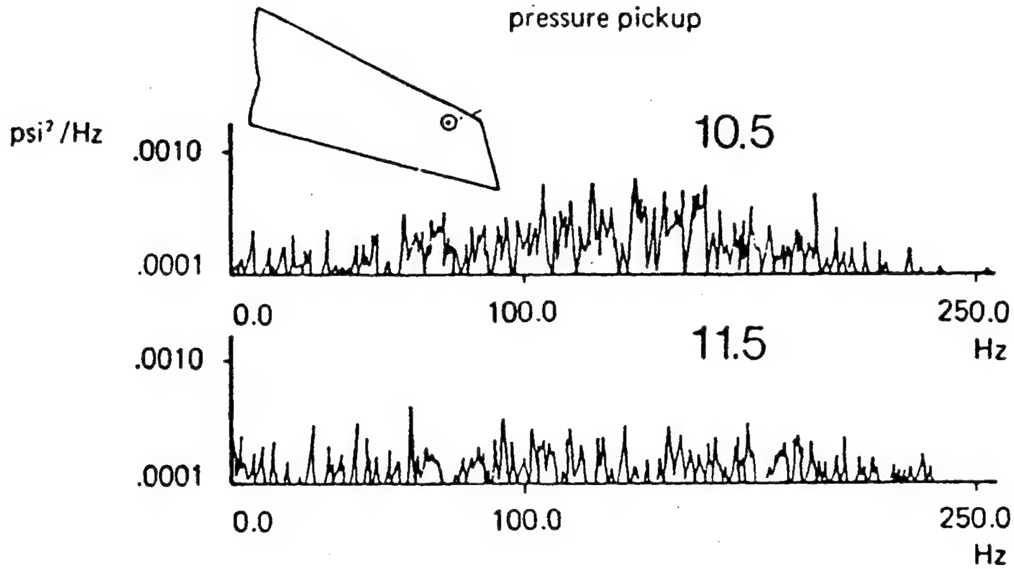


FIGURE 21. BUFFET PRESSURE POWER SPECTRAL DENSITIES
$M = 0.8$, $\Lambda = 25°$ [10]

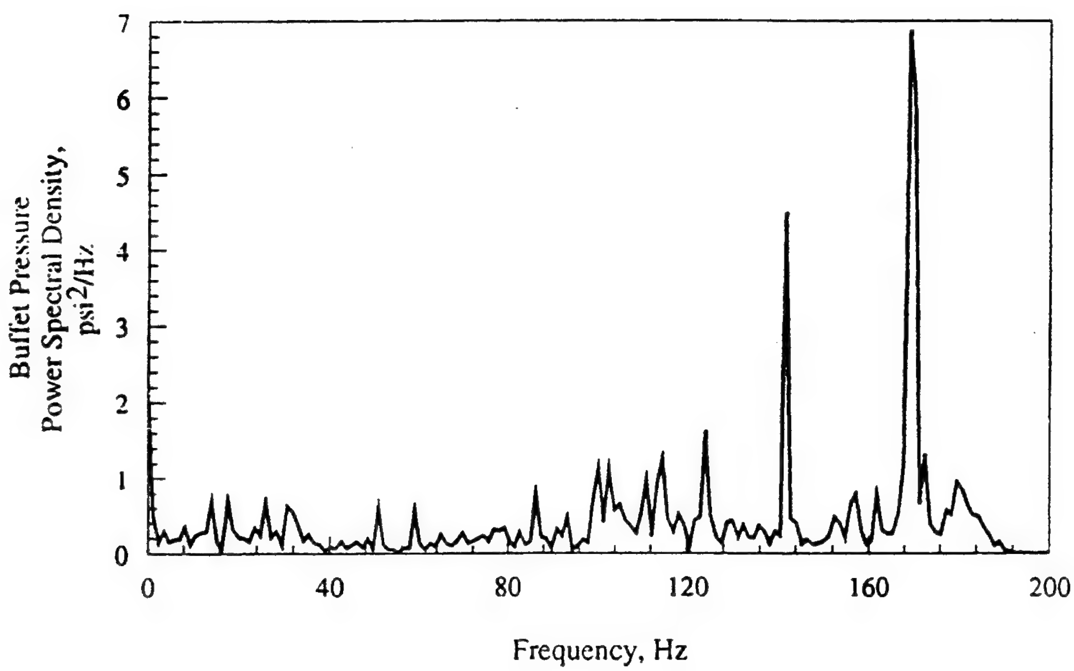


FIGURE 22. COMPUTED BUFFET PRESSURE PSD (RUN 10, POINT 5, CHANNEL 0)

## 3.2 AEROELASTIC RESULTS (ANALYTICAL).

Although it has been established that the validity of the estimated data is questionable, the analytical segment of the research (as detailed in section 2.3) was completed to establish the soundness of the overall methodology. Root bending moment, tip deflection, and tip twist were computed using the estimated PSD of the flight condition shown in figure 22; the bending moment results are summarized in figure 23. It was decided that it would not be prudent and may be potentially misleading to perform parameter studies.

For comparison, the bending moment was also computed using data from reference 5 and included previously as figure 6. As the buffet forcing function is nearly constant with frequency in the region of concern, a constant PSD equal to 0.0001 $psi^2/Hz$ was used for all transducers and frequencies. The results are summarized in figure 24.

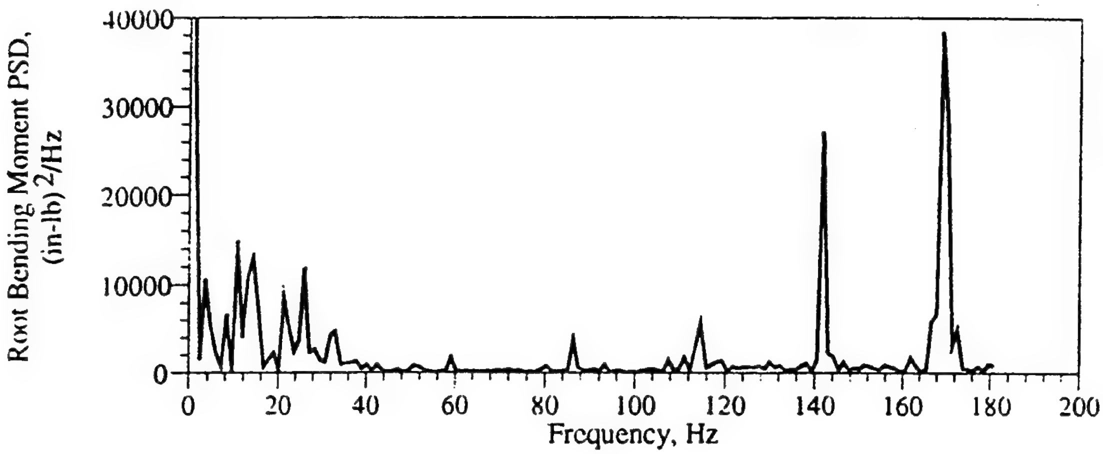


FIGURE 23. COMPUTED BENDING MOMENT FOR THE BEECH KING AIR 200 (RUN 10, POINT 5, LEFT SIDE)

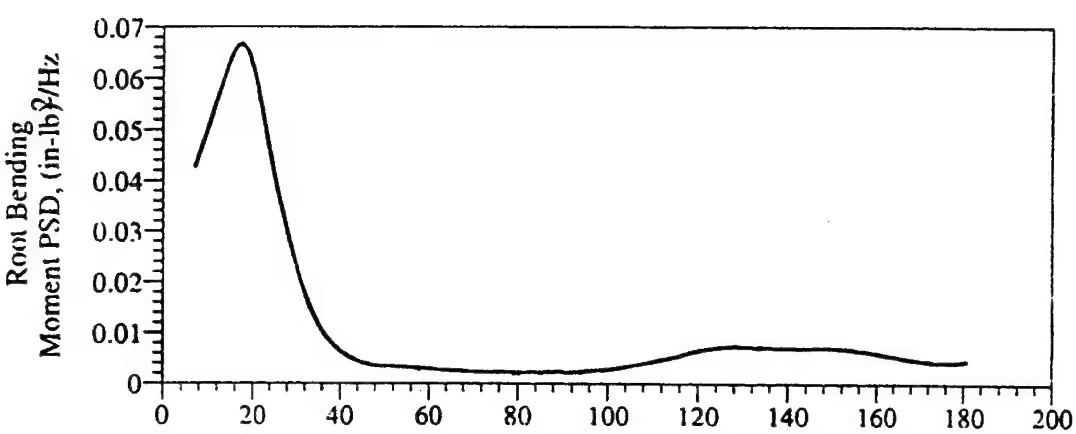


FIGURE 24. COMPUTED BENDING MOMENT FOR THE BEECH KING AIR 200 WITH BUFFET PRESSURE PSDS EQUAL $10^{-4}$

Comparing figures 23 and 24 further confirms that the current experimental data was not valid. The root bending moment shown in figure 23 is completely unreasonable, while the data shown in figure 24 is reasonable. This illustrates the importance of properly estimating the buffet forcing function. An example of a bending moment PSD from reference 3 is included as figure 25 for further comparison.

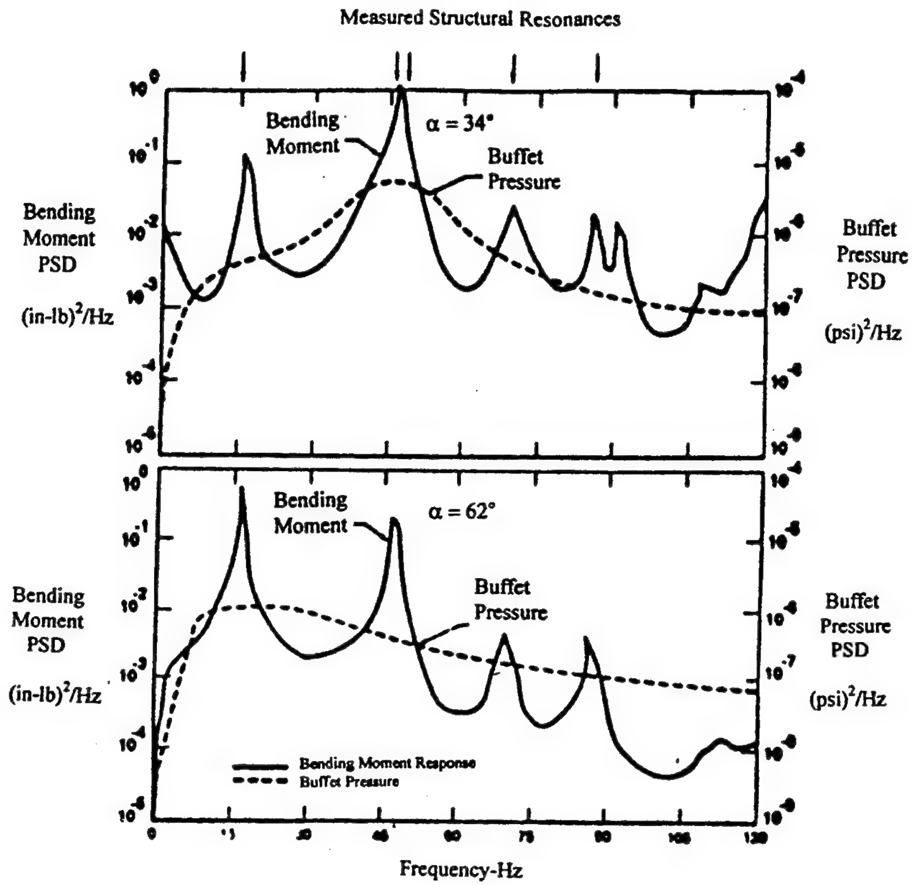


FIGURE 25. PREDICTED BENDING MOMENT PSDS [3] F-18 VERTICAL TAIL 12 PERCENT WIND TUNNEL MODEL V=81.5 FT/SEC

## 4. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK.

The most significant conclusion of this project is reaffirmation of the complex nature of the topic. Extreme care needs to be taken in each segment of the work. Experimental equipment should have known the characteristics prior to testing and should be operated by experienced personnel. The overall feasibility of the methodology has been proven through example and can be easily improved by reducing the number of modules.

The recommendations for future work are

- Repeat the experimental segment of this current work with a data acquisition system which has known characteristics and a built-in data reduction system. Ideally the tail would

contain a denser pressure transducer array to reduce the interpolation and extrapolation. This would allow the use of the tools already established to quickly establish significant parameters.

- After establishing the significant parameters, a much smaller test matrix could be implemented to perform scale studies. In particular several size models should be tested to establish the tunnel/model ratio and Reynold's effects.

- Dynamic testing should be performed to determine the effects of rates on the severity of buffet. This would include rate of change of angle of attack and heading angle. This testing is critical to a comprehensive study and subsequent modeling of buffet effect as the most severe tail buffet typically occurs during stall recovery. This flight condition is obviously very rate dependent. Buffet prediction based on the aforementioned dynamic testing would require additional analysis in postprocessing, as the result is nonstationary by definition.

- An additional study of interest would be the experimental testing of a model with simulated fuselage torsion. The model would ideally have a spring in the tail section that would allow the torsion mode of the fuselage to be excited. This study would analyze this effect on the severity of buffet.

- The analytical modeling should be improved to include convergence of both the structural and aerodynamic models. The structural model would be much improved with the inclusion of ground vibration data. This would insure accurate modelling. Many of the computations of this phase could also be easily implemented in ASTROS. This approach would render the modeling more synergistic and portable.

## 5. REFERENCES


1. Triplett, W. E., "Pressure Measurements on Twin Vertical Tails in Buffeting Flow," *Journal of Aircraft*, Vol. 20, No. 11, November 1983, pp. 920-925.

2. Mauk, C. S., "Methods of Stochastic Modeling of Antisymmetric Buffet Loads on Horizontal Stabilizers in Massively Separated Flows," Aerotech Technical Report AER TR93-052, May 5, 1993.

3. Ferman, M. A., Patel, S. R., Zimmerman, N. H., and Gerstenkorn, G., "A Unified Approach to Buffet Response of Fighter Aircraft Empennage," AGARD CP-483, 1990.

4. Butler, G. F., and Jones, J. G., "The Prediction of Buffeting Response in Flight from Wind-Tunnel Measurements on Models of Conventional Construction," *Aeronautical Journal*, Aug./Sept. 1984.

5. Lemley, C. E., and Mullans, R. E., "Buffeting Pressures on a Swept Wing in Transonic Flight - Comparison of Model and Full-Scale Measurements," MCAIR 73-005, 1973.

6. Bendat, J. S., and Piersol, A. G., Random Data: Analysis and Measurement Procedures, John Wiley & Sons, New York, 1986.

7. Neill, D. J., and Herendeen, D. L., Astros Enhancements: Volume I - Astros User's Manual,WL-TR-93-3025, March 1993.

8. Roskam, J., Airplane Design, Part III: Layout Design of Cockpit, Fuselage, Wing and Empennage: Cutaways and Inboard Profiles, Roskam Aviation and Engineering Corporation, Ottawa, KS, 1989.

9. Taylor, J. W. R., Jane's All The World Aircraft, Jane's Publishing Company, London,

10. Becker, J., and Dau, K., "Evaluation of Vibration Levels at the Pilot Seat Caused by Wing Flow Separation," AGARD CP-226, 1977.

# APPENDIX A

## Test Matrix
## and
## Test Conditions

The test matrix contains the columns:

| | |
|---|---|
| run - | run number |
| q - | dynamic pressure (psf) |
| qcor- | dynamic pressure with wind tunnel corrections |
| alp cor - | angle of attack with wind tunnel corrections, in degrees |
| psi - | heading angle in degrees |
| rudder - | rudder deflection in degrees |
| elevator - | elevator deflection in degrees |
| flap - | flap deployment (full is 35 degrees) |

The computed test conditions contains variables needed for other computations; the entries differing from the test matrix are:

| | |
|---|---|
| temp - | tunnel temperature in °F |
| baro - | barometric pressure in in. Hg |
| T - R - | temperature in Rankine |
| rho - | density in slugs per cubic foot |
| V - | velocity in ft/s |
| Vsound - | speed of sound in ft/s |
| Mach - | Mach number |
| Rn/ft | Reynolds number per foot |

# Test Matrix

| run | q | qcor | alp cor | psi | rudder | elevator | flap | |
|---|---|---|---|---|---|---|---|---|
| 1 | – | – | -- | 0 | 0 | 0 | 0 | noise floor |
| 2 | 25 | {25, 25.1, 25, 25.2, 25} | {0.4, 6.4, 11.5, 12.7, 13.7} | 0 | 0 | 0 | 0 | |
| 3 | 25 | {25, 25, 25.2, 25.2, 25.2, 25.1, | {0.2, 5.7, 10.9, 11.9, 13, 14, | 0 | 0 | 0 | 0 | |
| • | • | 25, 25.2, 25.4, 25.3, 25.2, 25.3, | 14.9, 16, 16.9, 18, 19.1, 20.1, | • | • | • | • | |
| • | • | 25.3, 25.4, 25.4, 25.5, 25.5, 25.5} | 21, 22, 23, 24.1, 25.1, 26} | • | • | • | • | |
| 4 | 25 | {24.9, 24.9, 24.9, 25.1, 25.4, 25.7} | {0.3, 5.9, 10.9, 16.1, 21.1, 26} | -10 | 0 | 0 | 0 | |
| 5 | 25 | {25, 25, 25, 25.1, 25.4} | {0.3, 5.8, 11, 16.1, 21.1} | -20 | 0 | 0 | 0 | |
| 6 | 40 | {40.9, 40.4, 41.1, 40.7} | {0.2, 5.8, 11, 16.1} | 0 | 0 | 0 | 0 | |
| 7 | 35 | {35.2, 35.3, 35.1, 35.1, 35.4, 35.7} | {0.2, 5.8, 11, 16, 21, 2.7} | 0 | 0 | 0 | 0 | |
| 8 | 35 | {34.7, 34.9, 35, 35.1, 35.5} | {0.3, 5.9, 11.1, 16.1, 21} | -10 | 0 | 0 | 0 | |
| 9 | 35 | {35.2, 35.3, 35.1, 35.5} | {0.3, 5.9, 11.1, 16.1} | -20 | 0 | 0 | 0 | |
| 10 | 25 | {25, 25.1, 25.3, 25.4, 25.5} | {0.2, 10.9, 16.1, 21, 26} | 0 | -10 | 0 | 0 | |
| 11 | 25 | {24.8, 24.9, 24.8, 25.2, 25.3} | {0.3, 5.7, 11.1, 16.2, 21} | -10 | -10 | 0 | 0 | |
| 12 | 25 | {24.9, 25, 25.1, 25.4, 25.3} | {0.3, 5.7, 11, 16, 21} | -20 | -10 | 0 | 0 | |
| 13 | 35 | {34.7, 35.2, 34.8, 35.5, 35.7} | {0.2, 11, 16, 21, 26} | 0 | -10 | 0 | 0 | |
| 14 | 35 | {35, 35, 35, 35, 35.4} | {0.2, 5.7, 11, 16, 21.1} | -10 | -10 | 0 | 0 | |
| 15 | 35 | {35.1, 34.8, 35, 35.1, 35.5} | {0.3, 5.8, 11.1, 16.1, 21} | -20 | -10 | 0 | 0 | |
| 16 | 25 | {24.8, 25, 25, 25.3, 25.6} | {0.2, 11.1, 16.1, 21.1, 26.1} | 0 | -20 | 0 | 0 | |
| 17 | 25 | {24.8, 24.8, 24.9, 25.1, 25.3} | {0.3, 5.9, 10.9, 16.2, 21} | -10 | -20 | 0 | 0 | |
| 18 | 25 | {25, 25, 25.1, 25.1, 25.4} | {0.3, 5.8, 10.9, 16.1, 21.1} | -20 | -20 | 0 | 0 | |
| 19 | 35 | {34.9, 34.8, 35.1, 35.2, 35.5, 36} | {0.2, 5.8, 11, 16, 21.1, 26} | 0 | -20 | 0 | 0 | |
| 20 | 35 | {34.8, 35, 34.9, 35, 35.5} | {0.3, 5.9, 11.1, 16, 21} | -10 | -20 | 0 | 0 | |
| 21 | 35 | {35, 35, 35, 35.4, 35.5} | {0.3, 5.8, 11, 16, 21.1} | -20 | -20 | 0 | 0 | |
| 22 | 25 | {25.2, 25.1, 25.1, 25.5, 25.6} | {0.2, 11.3, 16.4, 21.5, 26.5} | 0 | 0 | -20 | 0 | |
| 23 | 25 | {25.1, 25, 24.9, 25, 25.4} | {0.2, 5.9, 11.3, 16.5, 21.5} | -10 | 0 | -20 | 0 | |
| 24 | 25 | {25.4, 25.5, 25.5, 25.5, 25.1} | {0.4, 6, 11.2, 16.6, 21.5} | -20 | 0 | -20 | 0 | |
| 25 | 35 | {35.4, 35.5, 35.6, 36, 36} | {0.2, 11.3, 16.3, 21.4, 26.4} | 0 | 0 | -20 | 0 | |
| 26 | 35 | {35, 35, 35.2, 35.1, 35.4} | {0.2, 5.7, 10.9, 15.9, 20.9} | -10 | 0 | -20 | 0 | |
| 27 | 35 | {35.2, 35.3, 35.2, 35.2, 35.3} | {0.2, 5.6, 10.9, 15.9, 20.9} | -20 | 0 | -20 | 0 | |
| 28 | 25 | {24.9, 24.9, 25.2, 25.1, 25.4} | {0.1, 10.7, 16, 21, 26} | 0 | -10 | -20 | 0 | |
| 29 | 25 | {25.1, 25.1, 25.1, 25.1, 25.3} | {0.2, 5.6, 10.9, 16, 21} | -10 | -10 | -20 | 0 | |
| 30 | 25 | {25.2, 25.2, 25.1, 25.2, 25.2} | {0.2, 5.7, 10.7, 16, 21} | -20 | -10 | -20 | 0 | |
| 31 | 35 | {35.2, 35.1, 35.1, 35.5, 35.8} | {0.1, 11, 15.9, 21, 26} | 0 | -10 | -20 | 0 | |
| 32 | 35 | {35, 35, 35, 35.2, 35.5} | {0.2, 5.6, 10.8, 15.9, 20.9} | -10 | -10 | -20 | 0 | |
| 33 | 35 | {35, 35.1, 35.2, 35.2, 35.6} | {0.2, 5.6, 10.8, 16, 21} | -20 | -10 | -20 | 0 | |
| 34 | 35 | {34.7, 34.9, 35.3, 35.4, 35.6} | {0.1, 10.9, 16, 21, 26} | 0 | -20 | -20 | 0 | |
| 35 | 35 | {34.9, 34.9, 34.9, .35.1, 35.4} | {0.2, 5.7, 10.8, 16, 21} | -10 | -20 | -20 | 0 | |
| 36 | 35 | {35, 34.9, 35, 35.2, 35.5} | {0.2, 5.6, 11, 16, 21} | -20 | -20 | -20 | 0 | |
| 37 | 35 | {35.1, 34.9, 35.3, 35.6, 35.7} | {0.4, 11, 16.1, 21.1, 26} | 0 | 0 | 20 | 0 | |
| 38 | 35 | {34.9, 35.2, 35.2, 35.2, 35.6} | {0.4, 6, 11.1, 16.2, 21.1} | -10 | 0 | 20 | 0 | |
| 39 | 35 | {35, 35.3, 35.2, 35.3, 35.5} | {0.4, 5.9, 11, 16.1, 21.1} | -20 | 0 | 20 | 0 | |
| 40 | 35 | {34.5, 34.8, 35.1, 35.4, 35.8} | {0.4, 11.1, 16.1, 21.1, 26.1} | 0 | -10 | 20 | 0 | |
| 41 | 35 | {34.8, 34.8, 34.9, 35.1, 35.4} | {0.4, 6, 11.1, 16.2, 21} | -10 | -10 | 20 | 0 | |
| 42 | 35 | {35.3, 35.1, 35.3, 35.3, 35.3} | {0.4, 5.9, 11, 16.2, 21.1} | -20 | -10 | 20 | 0 | |
| 43 | 35 | {35, 35, 35.8, 35.5, 35.7} | {0.3, 11.1, 16.1, 21.1, 26.1} | 0 | -20 | 20 | 0 | |
| 44 | 35 | {34.9, 34.9, 35, 35.3, 35.6} | {0.4, 5.8, 11, 16.1, 21} | -10 | -20 | 20 | 0 | |
| 45 | 35 | {35, 34.9, 35.1, 35.2, 35.7} | {0.4, 5.8, 10.9, 16.1, 21.1} | -20 | -20 | 20 | 0 | |
| 46 | 35 | {34.8, 35.1, 35.3, 36} | {0.9, 11.7, 16.5, 21.4} | 0 | -20 | 20 | Full | |
| 47 | 35 | {35, 35.1, 35.3, 35.8} | {0.9, 11.6, 16.5, 21.4} | -10 | -20 | 20 | Full | |
| 48 | 35 | {35, 35.1, 35.5, 35.7} | {0.9, 11.4, 16.5, 21.3} | -20 | -20 | 20 | Full | |
| 49 | 35 | {34.8, 34.9, 35.5, 36} | {0.9, 11.7, 16.5, 21.4} | 0 | 0 | 20 | Full | |
| 50 | 35 | {35, 35, 35.4, 35.8} | {0.9, 11.6, 16.5, 21.4} | -10 | 0 | 20 | Full | |
| 51 | 35 | {35.2, 34.9, 35.3, 35.6} | {0.9, 11.4, 16.6, 21.3} | -20 | 0 | 20 | Full | |
| 52 | 35 | {34.9, 34.8, 35.2, 35.8} | {0.9, 11.6, 16.5, 21.4} | 0 | -10 | 20 | Full | |
| 53 | 35 | {35.2, 34.8, 35.2, 35.7} | {0.9, 11.5, 16.5, 21.3} | -10 | -10 | 20 | Full | |

# Test Matrix

| run | q | qcor | alp cor | psi | rudder | elevator | flap |
|---|---|---|---|---|---|---|---|
| 54 | 35 | {35.1, 35.1, 35.3, 35.7} | {0.9, 11.5, 16.5, 21.3} | -20 | -10 | 20 | Full |
| 55 | 35 | {35.3, 35.3, 35.4, 35.7} | {0.7, 11.5, 16.3, 21.3} | 0 | 0 | -20 | Full |
| 56 | 35 | {34.9, 35.1, 35.6, 35.8} | {0.7, 11.5, 16.4, 21.2} | -10 | 0 | -20 | Full |
| 57 | 35 | {35.2, 35.1, 35.3, 35.7} | {0.7, 11.3, 16.5, 21.5} | -20 | 0 | -20 | Full |
| 58 | 35 | {35, 35, 35.4, 35.8} | {0.7, 11.3, 16.3, 21.3] | 0 | -10 | -20 | Full |
| 59 | 35 | {34.8, 35.3, 35.4, 35.8} | {0.7, 11.4, 16.3, 21.3} | -10 | -10 | -20 | Full |
| 60 | 35 | {35.2, 35.1, 35.5, 35.7} | {0.7, 11.5, 16.4, 21.3} | -20 | -10 | -20 | Full |
| 61 | 35 | {34.8, 35.1, 35.3, 35.7} | {0.7, 11.3, 16.4, 21.3} | 0 | -20 | -20 | Full |
| 62 | 35 | {34.9, 35.3, 35.5, 35.8, 35.3, 35.1, 35.7, 35.7} | {0.7, 11.5, 16.3, 21.2, 0.7, 11.4, 16.4, 21.2} | -10 | -20 | -20 | Full |
| 63 | 35 | {34.7, 35, 35.3, 36} | {0.8, 11.5, 16.4, 21.3} | -20 | -20 | -20 | Full |
| 64 | 35 | {35, 35, 35.4, 35.4} | {0.8, 11.5, 16.4, 11.6} | 0 | 0 | 0 | Full |
| 65 | 35 | {34.9, 35.3, 35.4, 36.1} | {0.8, 11.4, 16.4, 21.4} | -10 | 0 | 0 | Full |
| 66 | 35 | {34.9, 35.2, 35.3, 35.6} | {0.8, 11.4, 16.4, 21.3} | -20 | 0 | 0 | Full |
| 67 | 35 | {35, 34.9, 35.4, 35.7} | {0.8, 11.5, 16.6, 21.3} | 0 | -10 | 0 | Full |
| 68 | 35 | {35, 35.1, 35.1} | {0.8, 11.3, 11.2} | -10 | -10 | 0 | Full |
| 69 | 25 | {24.8, 24.8} | {0, 0} | -20 | -10 | 0 | Full |
| 70 | -- | -- | -- | -- | -- | -- | -- noise floor |
| 71 | 35 | {35, 34.3, 35.3, 36.3} | {0.8, 11.6, 16.4, 21.3} | 0 | -20 | 0 | Full |
| 72 | 35 | {35.3, 35.2, 35.4, 35.1} | {0.8, 11.6, 16.4, 21.2} | -10 | -20 | 0 | Full |
| 73 | 35 | {35.3, 35.1, 35.4, 35.8} | {0.8, 11.4, 16.6, 21.3} | -20 | -20 | 0 | Full |
| 74 | 25 | {25.3, 25.1, 25.2, 25.5} | {0.7, 11.4, 16.3, 21.2} | 0 | 0 | -20 | Full |
| 75 | 41 | {40.9, 41.1, 41.3, 41.8} | {0.7, 11.4, 16.3, 21.2} | 0 | 0 | -20 | Full |

# Computed Test Conditions

| run | alpcor | qc | temp | baro | T-R | rho | mu | V | Vsound | Mach | Rn/ft | rho-ast | rho ave | Mave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0 | 0.0 | 73.57 | 28.43 | 533.24 | 2.197E-03 | 3.828E-07 | 0.000 | 1132.00 | 0.000 | 0.00E+00 | 1.06E-07 | | |
| 1 | 0.0 | 0.0 | 73.92 | 28.43 | 533.59 | 2.196E-03 | 3.830E-07 | 0.000 | 1132.37 | 0.000 | 0.00E+00 | 1.06E-07 | | |
| 2 | 0.4 | 25.0 | 73.57 | 28.40 | 533.24 | 2.194E-03 | 3.828E-07 | 150.950 | 1132.00 | 0.133 | 8.65E+05 | 1.06E-07 | | |
| 2 | 6.4 | 25.1 | 79.23 | 28.39 | 538.90 | 2.171E-03 | 3.860E-07 | 152.074 | 1137.99 | 0.134 | 8.55E+05 | 1.05E-07 | | |
| 2 | 11.5 | 25.0 | 81.17 | 28.39 | 540.84 | 2.163E-03 | 3.871E-07 | 152.034 | 1140.04 | 0.133 | 8.50E+05 | 1.04E-07 | 1.04E-07 | 0.134 |
| 2 | 12.7 | 25.2 | 82.94 | 28.39 | 542.61 | 2.156E-03 | 3.880E-07 | 152.890 | 1141.90 | 0.134 | 8.50E+05 | 1.04E-07 | | |
| 2 | 13.7 | 25.0 | 84.71 | 28.39 | 544.38 | 2.149E-03 | 3.890E-07 | 152.551 | 1143.76 | 0.133 | 8.42E+05 | 1.04E-07 | | |
| 3 | 0.2 | 25.0 | 76.57 | 28.37 | 536.24 | 2.180E-03 | 3.845E-07 | 151.449 | 1135.18 | 0.133 | 8.59E+05 | 1.05E-07 | | |
| 3 | 5.7 | 25.0 | 82.23 | 28.36 | 541.90 | 2.157E-03 | 3.876E-07 | 152.267 | 1141.16 | 0.133 | 8.47E+05 | 1.04E-07 | | |
| 3 | 10.9 | 25.2 | 84.53 | 28.37 | 544.20 | 2.148E-03 | 3.889E-07 | 153.188 | 1143.57 | 0.134 | 8.46E+05 | 1.04E-07 | | |
| 3 | 11.9 | 25.2 | 85.42 | 28.37 | 545.09 | 2.145E-03 | 3.894E-07 | 153.291 | 1144.50 | 0.134 | 8.44E+05 | 1.03E-07 | | |
| 3 | 13.0 | 25.2 | 86.65 | 28.38 | 546.32 | 2.140E-03 | 3.901E-07 | 153.455 | 1145.80 | 0.134 | 8.42E+05 | 1.03E-07 | | |
| 3 | 14.0 | 25.1 | 87.36 | 28.37 | 547.03 | 2.137E-03 | 3.905E-07 | 153.260 | 1146.54 | 0.134 | 8.39E+05 | 1.03E-07 | | |
| 3 | 14.9 | 25.0 | 88.78 | 28.37 | 548.45 | 2.131E-03 | 3.913E-07 | 153.173 | 1148.03 | 0.133 | 8.34E+05 | 1.03E-07 | | |
| 3 | 16.0 | 25.2 | 89.48 | 28.37 | 549.15 | 2.129E-03 | 3.917E-07 | 153.873 | 1148.77 | 0.134 | 8.36E+05 | 1.03E-07 | | |
| 3 | 16.9 | 25.4 | 89.13 | 28.38 | 548.80 | 2.131E-03 | 3.915E-07 | 154.400 | 1148.40 | 0.134 | 8.40E+05 | 1.03E-07 | 1.03E-07 | 0.134 |
| 3 | 18.0 | 25.3 | 90.90 | 28.36 | 550.57 | 2.123E-03 | 3.925E-07 | 154.398 | 1150.24 | 0.134 | 8.35E+05 | 1.02E-07 | | |
| 3 | 19.1 | 25.2 | 91.08 | 28.37 | 550.75 | 2.123E-03 | 3.926E-07 | 154.085 | 1150.43 | 0.134 | 8.33E+05 | 1.02E-07 | | |
| 3 | 20.1 | 25.3 | 91.78 | 28.36 | 551.45 | 2.119E-03 | 3.930E-07 | 154.522 | 1151.17 | 0.134 | 8.33E+05 | 1.02E-07 | | |
| 3 | 21.0 | 25.3 | 92.67 | 28.36 | 552.34 | 2.116E-03 | 3.935E-07 | 154.646 | 1152.09 | 0.134 | 8.32E+05 | 1.02E-07 | | |
| 3 | 22.0 | 25.4 | 93.20 | 28.35 | 552.87 | 2.113E-03 | 3.938E-07 | 155.047 | 1152.64 | 0.135 | 8.32E+05 | 1.02E-07 | | |
| 3 | 23.0 | 25.4 | 93.73 | 28.35 | 553.40 | 2.111E-03 | 3.941E-07 | 155.132 | 1153.20 | 0.135 | 8.31E+05 | 1.02E-07 | | |
| 3 | 24.1 | 25.5 | 94.44 | 28.35 | 554.11 | 2.108E-03 | 3.945E-07 | 155.526 | 1153.93 | 0.135 | 8.31E+05 | 1.02E-07 | | |
| 3 | 25.1 | 25.5 | 95.14 | 28.35 | 554.81 | 2.105E-03 | 3.949E-07 | 155.636 | 1154.67 | 0.135 | 8.30E+05 | 1.02E-07 | | |
| 3 | 26.0 | 25.5 | 95.32 | 28.35 | 554.99 | 2.105E-03 | 3.950E-07 | 155.660 | 1154.85 | 0.135 | 8.30E+05 | 1.02E-07 | | |
| 4 | 0.3 | 24.9 | 90.37 | 28.35 | 550.04 | 2.123E-03 | 3.922E-07 | 153.141 | 1149.69 | 0.133 | 8.29E+05 | 1.02E-07 | | |
| 4 | 5.9 | 24.9 | 92.67 | 28.35 | 552.34 | 2.115E-03 | 3.935E-07 | 153.461 | 1152.09 | 0.133 | 8.25E+05 | 1.02E-07 | | |
| 4 | 10.9 | 24.9 | 94.08 | 28.34 | 553.75 | 2.109E-03 | 3.943E-07 | 153.679 | 1153.56 | 0.133 | 8.22E+05 | 1.02E-07 | | |
| 4 | 16.1 | 25.1 | 96.03 | 28.34 | 555.70 | 2.102E-03 | 3.953E-07 | 154.555 | 1155.59 | 0.134 | 8.22E+05 | 1.01E-07 | 1.02E-07 | 0.134 |
| 4 | 21.1 | 25.4 | 96.56 | 28.34 | 556.23 | 2.100E-03 | 3.956E-07 | 155.550 | 1156.14 | 0.135 | 8.25E+05 | 1.01E-07 | | |
| 4 | 26.0 | 25.7 | 97.09 | 28.35 | 556.76 | 2.098E-03 | 3.959E-07 | 156.518 | 1156.69 | 0.135 | 8.29E+05 | 1.01E-07 | | |
| 5 | 0.3 | 25.0 | 97.62 | 28.35 | 557.29 | 2.096E-03 | 3.962E-07 | 154.456 | 1157.24 | 0.133 | 8.17E+05 | 1.01E-07 | | |
| 5 | 5.8 | 25.0 | 97.97 | 28.35 | 557.64 | 2.095E-03 | 3.964E-07 | 154.505 | 1157.61 | 0.133 | 8.16E+05 | 1.01E-07 | | |
| 5 | 11.0 | 25.0 | 98.50 | 28.34 | 558.17 | 2.092E-03 | 3.967E-07 | 154.590 | 1158.16 | 0.133 | 8.15E+05 | 1.01E-07 | 1.01E-07 | 0.134 |
| 5 | 16.1 | 25.1 | 98.33 | 28.35 | 558.00 | 2.093E-03 | 3.966E-07 | 154.863 | 1157.98 | 0.134 | 8.17E+05 | 1.01E-07 | | |
| 5 | 21.1 | 25.4 | 99.39 | 28.34 | 559.06 | 2.089E-03 | 3.972E-07 | 155.945 | 1159.08 | 0.135 | 8.20E+05 | 1.01E-07 | | |
| 6 | 0.2 | 40.9 | 82.23 | 28.40 | 541.90 | 2.160E-03 | 3.876E-07 | 194.624 | 1141.16 | 0.171 | 1.08E+06 | 1.04E-07 | | |
| 6 | 5.8 | 40.4 | 89.66 | 28.41 | 549.33 | 2.131E-03 | 3.918E-07 | 194.725 | 1148.95 | 0.169 | 1.06E+06 | 1.03E-07 | 1.03E-07 | 0.170 |
| 6 | 11.0 | 41.1 | 93.55 | 28.40 | 553.22 | 2.115E-03 | 3.940E-07 | 197.126 | 1153.01 | 0.171 | 1.06E+06 | 1.02E-07 | | |
| 6 | 16.1 | 40.7 | 96.91 | 28.41 | 556.58 | 2.103E-03 | 3.958E-07 | 196.732 | 1156.51 | 0.170 | 1.05E+06 | 1.01E-07 | | |
| 7 | 0.2 | 35.2 | 97.09 | 28.41 | 556.76 | 2.102E-03 | 3.959E-07 | 182.999 | 1156.69 | 0.158 | 9.72E+05 | 1.01E-07 | | |
| 7 | 5.8 | 35.3 | 97.80 | 28.41 | 557.47 | 2.100E-03 | 3.963E-07 | 183.375 | 1157.43 | 0.158 | 9.71E+05 | 1.01E-07 | | |
| 7 | 11.0 | 35.1 | 98.86 | 28.41 | 558.53 | 2.096E-03 | 3.969E-07 | 183.003 | 1158.53 | 0.158 | 9.66E+05 | 1.01E-07 | 1.01E-07 | 0.158 |
| 7 | 16.0 | 35.1 | 99.74 | 28.42 | 559.41 | 2.093E-03 | 3.974E-07 | 183.135 | 1159.44 | 0.158 | 9.65E+05 | 1.01E-07 | | |
| 7 | 21.0 | 35.4 | 101.16 | 28.43 | 560.83 | 2.089E-03 | 3.982E-07 | 184.111 | 1160.91 | 0.159 | 9.66E+05 | 1.01E-07 | | |
| 7 | 2.7 | 35.7 | 102.57 | 28.42 | 562.24 | 2.083E-03 | 3.990E-07 | 185.148 | 1162.37 | 0.159 | 9.67E+05 | 1.00E-07 | | |
| 8 | 0.3 | 34.7 | 103.28 | 28.42 | 562.95 | 2.080E-03 | 3.994E-07 | 182.664 | 1163.10 | 0.157 | 9.51E+05 | 1.00E-07 | | |
| 8 | 5.9 | 34.9 | 104.34 | 28.42 | 564.01 | 2.076E-03 | 3.999E-07 | 183.349 | 1164.20 | 0.157 | 9.52E+05 | 1.00E-07 | | |

## Computed Test Conditions

| run | alpcor | qc | temp | baro | T-R | rho | mu | V | Vsound | Mach | Rn/ft | rho-ast | rho ave | Mave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 11.1 | 35.0 | 104.34 | 28.41 | 564.01 | 2.076E-03 | 3.999E-07 | 183.637 | 1164.20 | 0.158 | 9.53E+05 | 1.00E-07 | 1.00E-07 | 0.158 |
| 8 | 16.1 | 35.1 | 105.22 | 28.42 | 564.89 | 2.073E-03 | 4.004E-07 | 184.031 | 1165.11 | 0.158 | 9.53E+05 | 1.00E-07 | | |
| 8 | 21.0 | 35.5 | 106.46 | 28.41 | 566.13 | 2.068E-03 | 4.011E-07 | 185.292 | 1166.39 | 0.159 | 9.55E+05 | 9.97E-08 | | |
| 9 | 0.3 | 35.2 | 107.52 | 28.41 | 567.19 | 2.064E-03 | 4.017E-07 | 184.680 | 1167.48 | 0.158 | 9.49E+05 | 9.95E-08 | | |
| 9 | 5.9 | 35.3 | 108.23 | 28.42 | 567.90 | 2.062E-03 | 4.021E-07 | 185.045 | 1168.21 | 0.158 | 9.49E+05 | 9.94E-08 | 9.95E-08 | 0.158 |
| 9 | 11.0 | 35.1 | 107.70 | 28.43 | 567.37 | 2.065E-03 | 4.018E-07 | 184.382 | 1167.66 | 0.158 | 9.48E+05 | 9.96E-08 | | |
| 9 | 16.1 | 35.5 | 108.94 | 28.43 | 568.61 | 2.060E-03 | 4.025E-07 | 185.658 | 1168.93 | 0.159 | 9.50E+05 | 9.93E-08 | | |
| 10 | 0.2 | 25.0 | 95.14 | 28.43 | 554.81 | 2.111E-03 | 3.949E-07 | 153.899 | 1154.67 | 0.133 | 8.23E+05 | 1.02E-07 | | |
| 10 | 10.9 | 25.1 | 98.68 | 28.43 | 558.35 | 2.098E-03 | 3.968E-07 | 154.687 | 1158.34 | 0.134 | 8.18E+05 | 1.01E-07 | | |
| 10 | 16.1 | 25.3 | 100.62 | 28.42 | 560.29 | 2.090E-03 | 3.979E-07 | 155.604 | 1160.36 | 0.134 | 8.17E+05 | 1.01E-07 | 1.01E-07 | 0.134 |
| 10 | 21.0 | 25.4 | 100.80 | 28.43 | 560.47 | 2.090E-03 | 3.980E-07 | 155.915 | 1160.54 | 0.134 | 8.19E+05 | 1.01E-07 | | |
| 10 | 26.0 | 25.5 | 101.86 | 28.41 | 561.53 | 2.085E-03 | 3.986E-07 | 156.412 | 1161.64 | 0.135 | 8.18E+05 | 1.01E-07 | | |
| 11 | 0.3 | 24.8 | 102.75 | 28.42 | 562.42 | 2.082E-03 | 3.991E-07 | 154.351 | 1162.56 | 0.133 | 8.05E+05 | 1.00E-07 | | |
| 11 | 5.7 | 24.9 | 102.39 | 28.42 | 562.06 | 2.084E-03 | 3.989E-07 | 154.602 | 1162.19 | 0.133 | 8.08E+05 | 1.00E-07 | | |
| 11 | 11.1 | 24.8 | 103.10 | 28.41 | 562.77 | 2.080E-03 | 3.993E-07 | 154.410 | 1162.92 | 0.133 | 8.05E+05 | 1.00E-07 | 1.00E-07 | 0.133 |
| 11 | 16.2 | 25.2 | 103.28 | 28.43 | 562.95 | 2.081E-03 | 3.994E-07 | 155.642 | 1163.10 | 0.134 | 8.11E+05 | 1.00E-07 | | |
| 11 | 21.0 | 25.3 | 103.81 | 28.42 | 563.48 | 2.078E-03 | 3.997E-07 | 156.035 | 1163.65 | 0.134 | 8.11E+05 | 1.00E-07 | | |
| 12 | 0.3 | 24.9 | 105.05 | 28.42 | 564.72 | 2.073E-03 | 4.003E-07 | 154.977 | 1164.93 | 0.133 | 8.03E+05 | 1.00E-07 | | |
| 12 | 5.7 | 25.0 | 105.22 | 28.42 | 564.89 | 2.073E-03 | 4.004E-07 | 155.302 | 1165.11 | 0.133 | 8.04E+05 | 1.00E-07 | | |
| 12 | 11.0 | 25.1 | 104.69 | 28.43 | 564.36 | 2.075E-03 | 4.001E-07 | 155.528 | 1164.56 | 0.134 | 8.07E+05 | 1.00E-07 | 1.00E-07 | 0.134 |
| 12 | 16.0 | 25.4 | 105.40 | 28.44 | 565.07 | 2.074E-03 | 4.005E-07 | 156.520 | 1165.29 | 0.134 | 8.10E+05 | 1.00E-07 | | |
| 12 | 21.0 | 25.3 | 105.58 | 28.43 | 565.25 | 2.072E-03 | 4.006E-07 | 156.258 | 1165.48 | 0.134 | 8.08E+05 | 9.99E-08 | | |
| 13 | 0.2 | 34.7 | 99.92 | 28.43 | 559.59 | 2.093E-03 | 3.975E-07 | 182.080 | 1159.63 | 0.157 | 9.59E+05 | 1.01E-07 | | |
| 13 | 11.0 | 35.2 | 106.11 | 28.42 | 565.78 | 2.070E-03 | 4.009E-07 | 184.437 | 1166.02 | 0.158 | 9.52E+05 | 9.98E-08 | | |
| 13 | 16.0 | 34.8 | 110.17 | 28.43 | 569.84 | 2.056E-03 | 4.032E-07 | 183.993 | 1170.21 | 0.157 | 9.38E+05 | 9.91E-08 | 9.95E-08 | 0.158 |
| 13 | 21.0 | 35.5 | 111.41 | 28.44 | 571.08 | 2.052E-03 | 4.038E-07 | 186.023 | 1171.48 | 0.159 | 9.45E+05 | 9.89E-08 | | |
| 13 | 26.0 | 35.7 | 112.47 | 28.44 | 572.14 | 2.049E-03 | 4.044E-07 | 186.694 | 1172.56 | 0.159 | 9.46E+05 | 9.88E-08 | | |
| 14 | 0.2 | 35.0 | 114.24 | 28.44 | 573.91 | 2.042E-03 | 4.054E-07 | 185.165 | 1174.38 | 0.158 | 9.33E+05 | 9.85E-08 | | |
| 14 | 5.7 | 35.0 | 114.95 | 28.43 | 574.62 | 2.039E-03 | 4.058E-07 | 185.305 | 1175.10 | 0.158 | 9.31E+05 | 9.83E-08 | | |
| 14 | 11.0 | 35.0 | 115.48 | 28.43 | 575.15 | 2.037E-03 | 4.061E-07 | 185.391 | 1175.64 | 0.158 | 9.30E+05 | 9.82E-08 | 9.82E-08 | 0.158 |
| 14 | 16.0 | 35.0 | 115.66 | 28.44 | 575.33 | 2.037E-03 | 4.061E-07 | 185.394 | 1175.82 | 0.158 | 9.30E+05 | 9.82E-08 | | |
| 14 | 21.1 | 35.4 | 116.72 | 28.44 | 576.39 | 2.033E-03 | 4.067E-07 | 186.622 | 1176.91 | 0.159 | 9.33E+05 | 9.80E-08 | | |
| 15 | 0.3 | 35.1 | 106.46 | 28.44 | 566.13 | 2.070E-03 | 4.011E-07 | 184.168 | 1166.39 | 0.158 | 9.50E+05 | 9.98E-08 | | |
| 15 | 5.8 | 34.8 | 110.53 | 28.43 | 570.20 | 2.055E-03 | 4.033E-07 | 184.050 | 1170.57 | 0.157 | 9.38E+05 | 9.91E-08 | | |
| 15 | 11.1 | 35.0 | 112.65 | 28.44 | 572.32 | 2.048E-03 | 4.045E-07 | 184.896 | 1172.75 | 0.158 | 9.36E+05 | 9.87E-08 | 9.89E-08 | 0.158 |
| 15 | 16.1 | 35.1 | 114.07 | 28.44 | 573.74 | 2.042E-03 | 4.053E-07 | 185.401 | 1174.19 | 0.158 | 9.34E+05 | 9.85E-08 | | |
| 15 | 21.0 | 35.5 | 116.01 | 28.45 | 575.68 | 2.036E-03 | 4.063E-07 | 186.732 | 1176.18 | 0.159 | 9.36E+05 | 9.82E-08 | | |
| 16 | 0.2 | 24.8 | 96.20 | 28.41 | 555.87 | 2.106E-03 | 3.954E-07 | 153.482 | 1155.77 | 0.133 | 8.17E+05 | 1.02E-07 | | |
| 16 | 11.1 | 25.0 | 101.33 | 28.41 | 561.00 | 2.086E-03 | 3.983E-07 | 154.809 | 1161.09 | 0.133 | 8.11E+05 | 1.01E-07 | | |
| 16 | 16.1 | 25.0 | 103.63 | 28.41 | 563.30 | 2.078E-03 | 3.996E-07 | 155.115 | 1163.47 | 0.133 | 8.07E+05 | 1.00E-07 | 1.00E-07 | 0.134 |
| 16 | 21.1 | 25.3 | 105.05 | 28.42 | 564.72 | 2.073E-03 | 4.003E-07 | 156.217 | 1164.93 | 0.134 | 8.09E+05 | 1.00E-07 | | |
| 16 | 26.1 | 25.6 | 106.28 | 28.41 | 565.95 | 2.069E-03 | 4.010E-07 | 157.324 | 1166.21 | 0.135 | 8.12E+05 | 9.98E-08 | | |
| 17 | 0.3 | 24.8 | 108.05 | 28.41 | 567.72 | 2.062E-03 | 4.020E-07 | 155.099 | 1168.03 | 0.133 | 7.96E+05 | 9.94E-08 | | |
| 17 | 5.9 | 24.8 | 108.58 | 28.40 | 568.25 | 2.059E-03 | 4.023E-07 | 155.192 | 1168.57 | 0.133 | 7.94E+05 | 9.93E-08 | | |
| 17 | 10.9 | 24.9 | 108.94 | 28.41 | 568.61 | 2.058E-03 | 4.025E-07 | 155.543 | 1168.93 | 0.133 | 7.96E+05 | 9.93E-08 | 9.93E-08 | 0.133 |
| 17 | 16.2 | 25.1 | 109.47 | 28.41 | 569.14 | 2.056E-03 | 4.028E-07 | 156.239 | 1169.48 | 0.134 | 7.98E+05 | 9.92E-08 | | |
| 17 | 21.0 | 25.3 | 110.00 | 28.40 | 569.67 | 2.054E-03 | 4.031E-07 | 156.944 | 1170.03 | 0.134 | 8.00E+05 | 9.91E-08 | | |

# Computed Test Conditions

| run | alpcor | qc | temp | baro | T-R | rho | mu | V | Vsound | Mach | Rn/ft | rho-ast | rho ave | Mave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 0.3 | 25.0 | 110.88 | 28.40 | 570.55 | 2.051E-03 | 4.035E-07 | 156.132 | 1170.93 | 0.133 | 7.94E+05 | 9.89E-08 | | |
| 18 | 5.8 | 25.0 | 111.24 | 28.40 | 570.91 | 2.050E-03 | 4.037E-07 | 156.180 | 1171.30 | 0.133 | 7.93E+05 | 9.89E-08 | | |
| 18 | 10.9 | 25.1 | 111.41 | 28.41 | 571.08 | 2.049E-03 | 4.038E-07 | 156.506 | 1171.48 | 0.134 | 7.94E+05 | 9.88E-08 | 9.88E-08 | 0.134 |
| 18 | 16.1 | 25.1 | 111.59 | 28.41 | 571.26 | 2.049E-03 | 4.039E-07 | 156.519 | 1171.66 | 0.134 | 7.94E+05 | 9.88E-08 | | |
| 18 | 21.1 | 25.4 | 112.30 | 28.41 | 571.97 | 2.047E-03 | 4.043E-07 | 157.549 | 1172.38 | 0.134 | 7.97E+05 | 9.87E-08 | | |
| 19 | 0.2 | 34.9 | 110.00 | 28.39 | 569.67 | 2.054E-03 | 4.031E-07 | 184.356 | 1170.03 | 0.158 | 9.39E+05 | 9.90E-08 | | |
| 19 | 5.8 | 34.8 | 113.89 | 28.40 | 573.56 | 2.040E-03 | 4.052E-07 | 184.707 | 1174.01 | 0.157 | 9.30E+05 | 9.84E-08 | | |
| 19 | 11.0 | 35.1 | 115.83 | 28.39 | 575.50 | 2.033E-03 | 4.062E-07 | 185.828 | 1176.00 | 0.158 | 9.30E+05 | 9.80E-08 | | |
| 19 | 16.0 | 35.2 | 117.25 | 28.39 | 576.92 | 2.028E-03 | 4.070E-07 | 186.334 | 1177.45 | 0.158 | 9.28E+05 | 9.78E-08 | 9.80E-08 | 0.158 |
| 19 | 21.1 | 35.5 | 118.49 | 28.39 | 578.16 | 2.024E-03 | 4.077E-07 | 187.314 | 1178.71 | 0.159 | 9.30E+05 | 9.76E-08 | | |
| 19 | 26.0 | 36.0 | 120.08 | 28.39 | 579.75 | 2.018E-03 | 4.086E-07 | 188.901 | 1180.33 | 0.160 | 9.33E+05 | 9.73E-08 | | |
| 20 | 0.3 | 34.8 | 104.69 | 28.38 | 564.36 | 2.072E-03 | 4.001E-07 | 183.271 | 1164.56 | 0.157 | 9.49E+05 | 9.99E-08 | | |
| 20 | 5.9 | 35.0 | 109.64 | 28.40 | 569.31 | 2.055E-03 | 4.029E-07 | 184.550 | 1169.66 | 0.158 | 9.42E+05 | 9.91E-08 | | |
| 20 | 11.1 | 34.9 | 112.47 | 28.39 | 572.14 | 2.044E-03 | 4.044E-07 | 184.782 | 1172.56 | 0.158 | 9.34E+05 | 9.86E-08 | 9.87E-08 | 0.158 |
| 20 | 16.0 | 35.0 | 114.60 | 28.38 | 574.27 | 2.036E-03 | 4.056E-07 | 185.402 | 1174.74 | 0.158 | 9.31E+05 | 9.82E-08 | | |
| 20 | 21.0 | 35.5 | 116.54 | 28.39 | 576.21 | 2.030E-03 | 4.066E-07 | 187.012 | 1176.73 | 0.159 | 9.34E+05 | 9.79E-08 | | |
| 21 | 0.3 | 35.0 | 118.66 | 28.39 | 578.33 | 2.023E-03 | 4.078E-07 | 186.032 | 1178.89 | 0.158 | 9.23E+05 | 9.75E-08 | | |
| 21 | 5.8 | 35.0 | 118.84 | 28.39 | 578.51 | 2.022E-03 | 4.079E-07 | 186.073 | 1179.07 | 0.158 | 9.22E+05 | 9.75E-08 | | |
| 21 | 11.0 | 35.0 | 120.08 | 28.39 | 579.75 | 2.018E-03 | 4.086E-07 | 186.259 | 1180.33 | 0.158 | 9.20E+05 | 9.73E-08 | 9.73E-08 | 0.158 |
| 21 | 16.0 | 35.4 | 120.61 | 28.39 | 580.28 | 2.016E-03 | 4.088E-07 | 187.419 | 1180.87 | 0.159 | 9.24E+05 | 9.72E-08 | | |
| 21 | 21.1 | 35.5 | 122.02 | 28.38 | 581.69 | 2.010E-03 | 4.096E-07 | 187.939 | 1182.31 | 0.159 | 9.22E+05 | 9.69E-08 | | |
| 22 | 0.2 | 25.2 | 81.88 | 28.53 | 541.55 | 2.171E-03 | 3.874E-07 | 152.371 | 1140.78 | 0.134 | 8.54E+05 | 1.05E-07 | | |
| 22 | 11.3 | 25.1 | 88.07 | 28.54 | 547.74 | 2.147E-03 | 3.909E-07 | 152.914 | 1147.28 | 0.133 | 8.40E+05 | 1.04E-07 | | |
| 22 | 16.4 | 25.1 | 90.90 | 28.53 | 550.57 | 2.135E-03 | 3.925E-07 | 153.329 | 1150.24 | 0.133 | 8.34E+05 | 1.03E-07 | 1.03E-07 | 0.134 |
| 22 | 21.5 | 25.5 | 92.49 | 28.53 | 552.16 | 2.129E-03 | 3.934E-07 | 154.770 | 1151.91 | 0.134 | 8.38E+05 | 1.03E-07 | | |
| 22 | 26.5 | 25.6 | 92.31 | 28.54 | 551.98 | 2.130E-03 | 3.933E-07 | 155.037 | 1151.72 | 0.135 | 8.40E+05 | 1.03E-07 | | |
| 23 | 0.2 | 25.1 | 95.67 | 28.54 | 555.34 | 2.117E-03 | 3.952E-07 | 153.982 | 1155.22 | 0.133 | 8.25E+05 | 1.02E-07 | | |
| 23 | 5.9 | 25.0 | 96.38 | 28.54 | 556.05 | 2.115E-03 | 3.955E-07 | 153.762 | 1155.96 | 0.133 | 8.22E+05 | 1.02E-07 | | |
| 23 | 11.3 | 24.9 | 96.56 | 28.54 | 556.23 | 2.114E-03 | 3.956E-07 | 153.479 | 1156.14 | 0.133 | 8.20E+05 | 1.02E-07 | 1.02E-07 | 0.133 |
| 23 | 16.5 | 25.0 | 96.20 | 28.54 | 555.87 | 2.115E-03 | 3.954E-07 | 153.749 | 1155.77 | 0.133 | 8.22E+05 | 1.02E-07 | | |
| 23 | 21.5 | 25.4 | 95.50 | 28.54 | 555.17 | 2.118E-03 | 3.951E-07 | 154.854 | 1155.04 | 0.134 | 8.30E+05 | 1.02E-07 | | |
| 24 | 0.4 | 25.4 | 99.56 | 28.54 | 559.23 | 2.103E-03 | 3.973E-07 | 155.431 | 1159.26 | 0.134 | 8.23E+05 | 1.01E-07 | | |
| 24 | 6.0 | 25.5 | 100.09 | 28.54 | 559.76 | 2.101E-03 | 3.976E-07 | 155.799 | 1159.81 | 0.134 | 8.23E+05 | 1.01E-07 | | |
| 24 | 11.2 | 25.5 | 100.62 | 28.54 | 560.29 | 2.099E-03 | 3.979E-07 | 155.873 | 1160.36 | 0.134 | 8.22E+05 | 1.01E-07 | 1.01E-07 | 0.134 |
| 24 | 16.6 | 25.5 | 100.62 | 28.55 | 560.29 | 2.099E-03 | 3.979E-07 | 155.862 | 1160.36 | 0.134 | 8.22E+05 | 1.01E-07 | | |
| 24 | 21.5 | 25.1 | 100.09 | 28.54 | 559.76 | 2.101E-03 | 3.976E-07 | 154.573 | 1159.81 | 0.133 | 8.17E+05 | 1.01E-07 | | |
| 25 | 0.2 | 35.4 | 97.44 | 28.55 | 557.11 | 2.112E-03 | 3.961E-07 | 183.107 | 1157.06 | 0.158 | 9.76E+05 | 1.02E-07 | | |
| 25 | 11.3 | 35.5 | 102.57 | 28.56 | 562.24 | 2.093E-03 | 3.990E-07 | 184.195 | 1162.37 | 0.158 | 9.66E+05 | 1.01E-07 | | |
| 25 | 16.3 | 35.6 | 103.98 | 28.56 | 563.65 | 2.088E-03 | 3.997E-07 | 184.674 | 1163.83 | 0.159 | 9.64E+05 | 1.01E-07 | 1.01E-07 | 0.159 |
| 25 | 21.4 | 36.0 | 105.22 | 28.57 | 564.89 | 2.084E-03 | 4.004E-07 | 185.874 | 1165.11 | 0.160 | 9.67E+05 | 1.01E-07 | | |
| 25 | 26.4 | 36.0 | 105.58 | 28.57 | 565.25 | 2.082E-03 | 4.006E-07 | 185.945 | 1165.48 | 0.160 | 9.67E+05 | 1.00E-07 | | |
| 26 | 0.2 | 35.0 | 95.50 | 28.59 | 555.17 | 2.122E-03 | 3.951E-07 | 181.639 | 1155.04 | 0.157 | 9.76E+05 | 1.02E-07 | | |
| 26 | 5.7 | 35.0 | 99.92 | 28.60 | 559.59 | 2.106E-03 | 3.975E-07 | 182.323 | 1159.63 | 0.157 | 9.66E+05 | 1.02E-07 | | |
| 26 | 10.9 | 35.2 | 103.28 | 28.59 | 562.95 | 2.093E-03 | 3.994E-07 | 183.404 | 1163.10 | 0.158 | 9.61E+05 | 1.01E-07 | 1.01E-07 | 0.158 |
| 26 | 15.9 | 35.1 | 103.45 | 28.60 | 563.12 | 2.093E-03 | 3.995E-07 | 183.160 | 1163.29 | 0.157 | 9.59E+05 | 1.01E-07 | | |
| 26 | 20.9 | 35.4 | 104.52 | 28.60 | 564.19 | 2.089E-03 | 4.000E-07 | 184.114 | 1164.38 | 0.158 | 9.61E+05 | 1.01E-07 | | |

# Computed Test Conditions

| run | alpcor | qc | temp | baro | T-R | rho | mu | V | Vsound | Mach | Rn/ft | rho-ast | rho ave | Mave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 0.2 | 35.2 | 109.11 | 28.59 | 568.78 | 2.071E-03 | 4.026E-07 | 184.352 | 1169.12 | 0.158 | 9.49E+05 | 9.99E-08 | | |
| 27 | 5.6 | 35.3 | 110.00 | 28.59 | 569.67 | 2.068E-03 | 4.031E-07 | 184.758 | 1170.03 | 0.158 | 9.48E+05 | 9.97E-08 | | |
| 27 | 10.9 | 35.2 | 110.71 | 28.60 | 570.38 | 2.066E-03 | 4.034E-07 | 184.597 | 1170.75 | 0.158 | 9.45E+05 | 9.96E-08 | 9.97E-08 | 0.158 |
| 27 | 15.9 | 35.2 | 111.06 | 28.60 | 570.73 | 2.065E-03 | 4.036E-07 | 184.642 | 1171.11 | 0.158 | 9.45E+05 | 9.96E-08 | | |
| 27 | 20.9 | 35.3 | 110.35 | 28.60 | 570.02 | 2.067E-03 | 4.032E-07 | 184.802 | 1170.39 | 0.158 | 9.47E+05 | 9.97E-08 | | |
| 28 | 0.1 | 24.9 | 102.57 | 28.59 | 562.24 | 2.095E-03 | 3.990E-07 | 154.179 | 1162.37 | 0.133 | 8.10E+05 | 1.01E-07 | | |
| 28 | 10.7 | 24.9 | 105.05 | 28.59 | 564.72 | 2.086E-03 | 4.003E-07 | 154.507 | 1164.93 | 0.133 | 8.05E+05 | 1.01E-07 | | |
| 28 | 16.0 | 25.2 | 106.64 | 28.59 | 566.31 | 2.081E-03 | 4.012E-07 | 155.643 | 1166.57 | 0.133 | 8.07E+05 | 1.00E-07 | 1.01E-07 | 0.133 |
| 28 | 21.0 | 25.1 | 106.64 | 28.59 | 566.31 | 2.080E-03 | 4.012E-07 | 155.345 | 1166.57 | 0.133 | 8.05E+05 | 1.00E-07 | | |
| 28 | 26.0 | 25.4 | 105.75 | 28.59 | 565.42 | 2.083E-03 | 4.007E-07 | 156.148 | 1165.66 | 0.134 | 8.12E+05 | 1.00E-07 | | |
| 29 | 0.2 | 25.1 | 108.58 | 28.59 | 568.25 | 2.073E-03 | 4.023E-07 | 155.611 | 1168.57 | 0.133 | 8.02E+05 | 1.00E-07 | | |
| 29 | 5.6 | 25.1 | 108.76 | 28.59 | 568.43 | 2.073E-03 | 4.024E-07 | 155.625 | 1168.75 | 0.133 | 8.02E+05 | 1.00E-07 | | |
| 29 | 10.9 | 25.1 | 108.58 | 28.59 | 568.25 | 2.073E-03 | 4.023E-07 | 155.611 | 1168.57 | 0.133 | 8.02E+05 | 1.00E-07 | 1.00E-07 | 0.133 |
| 29 | 16.0 | 25.1 | 109.64 | 28.59 | 569.31 | 2.070E-03 | 4.029E-07 | 155.746 | 1169.66 | 0.133 | 8.00E+05 | 9.98E-08 | | |
| 29 | 21.0 | 25.3 | 108.05 | 28.59 | 567.72 | 2.075E-03 | 4.020E-07 | 156.157 | 1168.03 | 0.134 | 8.06E+05 | 1.00E-07 | | |
| 30 | 0.2 | 25.2 | 110.35 | 28.59 | 570.02 | 2.066E-03 | 4.032E-07 | 156.174 | 1170.39 | 0.133 | 8.00E+05 | 9.97E-08 | | |
| 30 | 5.7 | 25.2 | 110.88 | 28.59 | 570.55 | 2.064E-03 | 4.035E-07 | 156.247 | 1170.93 | 0.133 | 7.99E+05 | 9.96E-08 | | |
| 30 | 10.7 | 25.1 | 110.71 | 28.58 | 570.38 | 2.065E-03 | 4.034E-07 | 155.923 | 1170.75 | 0.133 | 7.98E+05 | 9.96E-08 | 9.96E-08 | 0.133 |
| 30 | 16.0 | 25.2 | 111.06 | 28.59 | 570.73 | 2.064E-03 | 4.036E-07 | 156.271 | 1171.11 | 0.133 | 7.99E+05 | 9.95E-08 | | |
| 30 | 21.0 | 25.2 | 111.41 | 28.58 | 571.08 | 2.062E-03 | 4.038E-07 | 156.330 | 1171.48 | 0.133 | 7.98E+05 | 9.95E-08 | | |
| 31 | 0.1 | 35.2 | 98.68 | 28.59 | 558.35 | 2.110E-03 | 3.968E-07 | 182.679 | 1158.34 | 0.158 | 9.71E+05 | 1.02E-07 | | |
| 31 | 11.0 | 35.1 | 106.64 | 28.59 | 566.31 | 2.080E-03 | 4.012E-07 | 183.702 | 1166.57 | 0.157 | 9.52E+05 | 1.00E-07 | | |
| 31 | 15.9 | 35.1 | 109.11 | 28.58 | 568.78 | 2.070E-03 | 4.026E-07 | 184.141 | 1169.12 | 0.158 | 9.47E+05 | 9.98E-08 | 1.00E-07 | 0.158 |
| 31 | 21.0 | 35.5 | 111.24 | 28.57 | 570.91 | 2.062E-03 | 4.037E-07 | 185.545 | 1171.30 | 0.158 | 9.48E+05 | 9.95E-08 | | |
| 31 | 26.0 | 35.8 | 112.65 | 28.57 | 572.32 | 2.057E-03 | 4.045E-07 | 186.558 | 1172.75 | 0.159 | 9.49E+05 | 9.92E-08 | | |
| 32 | 0.2 | 35.0 | 115.66 | 28.57 | 575.33 | 2.046E-03 | 4.061E-07 | 184.959 | 1175.82 | 0.157 | 9.32E+05 | 9.87E-08 | | |
| 32 | 5.6 | 35.0 | 116.89 | 28.57 | 576.56 | 2.042E-03 | 4.068E-07 | 185.158 | 1177.09 | 0.157 | 9.29E+05 | 9.85E-08 | | |
| 32 | 10.8 | 35.0 | 117.25 | 28.56 | 576.92 | 2.040E-03 | 4.070E-07 | 185.240 | 1177.45 | 0.157 | 9.28E+05 | 9.84E-08 | 9.85E-08 | 0.158 |
| 32 | 15.9 | 35.2 | 116.54 | 28.57 | 576.21 | 2.043E-03 | 4.066E-07 | 185.629 | 1176.73 | 0.158 | 9.33E+05 | 9.85E-08 | | |
| 32 | 20.9 | 35.5 | 116.36 | 28.57 | 576.03 | 2.043E-03 | 4.065E-07 | 186.402 | 1176.54 | 0.158 | 9.37E+05 | 9.85E-08 | | |
| 33 | 0.2 | 35.0 | 112.12 | 28.54 | 571.79 | 2.057E-03 | 4.042E-07 | 184.491 | 1172.20 | 0.157 | 9.39E+05 | 9.92E-08 | | |
| 33 | 5.6 | 35.1 | 115.13 | 28.55 | 574.80 | 2.047E-03 | 4.059E-07 | 185.201 | 1175.28 | 0.158 | 9.34E+05 | 9.87E-08 | | |
| 33 | 10.8 | 35.2 | 117.43 | 28.55 | 577.10 | 2.039E-03 | 4.071E-07 | 185.835 | 1177.63 | 0.158 | 9.31E+05 | 9.83E-08 | 9.85E-08 | 0.158 |
| 33 | 16.0 | 35.2 | 118.13 | 28.55 | 577.80 | 2.036E-03 | 4.075E-07 | 185.962 | 1178.35 | 0.158 | 9.29E+05 | 9.82E-08 | | |
| 33 | 21.0 | 35.6 | 118.84 | 28.55 | 578.51 | 2.034E-03 | 4.079E-07 | 187.117 | 1179.07 | 0.159 | 9.33E+05 | 9.81E-08 | | |
| 34 | 0.1 | 34.7 | 112.65 | 28.54 | 572.32 | 2.055E-03 | 4.045E-07 | 183.771 | 1172.75 | 0.157 | 9.34E+05 | 9.91E-08 | | |
| 34 | 10.9 | 34.9 | 116.54 | 28.55 | 576.21 | 2.041E-03 | 4.066E-07 | 184.913 | 1176.73 | 0.157 | 9.28E+05 | 9.84E-08 | | |
| 34 | 16.0 | 35.3 | 117.25 | 28.54 | 576.92 | 2.039E-03 | 4.070E-07 | 186.096 | 1177.45 | 0.158 | 9.32E+05 | 9.83E-08 | 9.83E-08 | 0.158 |
| 34 | 21.0 | 35.4 | 119.72 | 28.55 | 579.39 | 2.030E-03 | 4.084E-07 | 186.746 | 1179.97 | 0.158 | 9.28E+05 | 9.79E-08 | | |
| 34 | 26.0 | 35.6 | 120.43 | 28.54 | 580.10 | 2.027E-03 | 4.087E-07 | 187.400 | 1180.69 | 0.159 | 9.30E+05 | 9.78E-08 | | |
| 35 | 0.2 | 34.9 | 123.26 | 28.55 | 582.93 | 2.018E-03 | 4.103E-07 | 185.988 | 1183.57 | 0.157 | 9.15E+05 | 9.73E-08 | | |
| 35 | 5.7 | 34.9 | 123.97 | 28.54 | 583.64 | 2.015E-03 | 4.107E-07 | 186.113 | 1184.29 | 0.157 | 9.13E+05 | 9.72E-08 | | |
| 35 | 10.8 | 34.9 | 124.50 | 28.54 | 584.17 | 2.013E-03 | 4.110E-07 | 186.224 | 1184.82 | 0.157 | 9.12E+05 | 9.71E-08 | 9.72E-08 | 0.157 |
| 35 | 16.0 | 35.1 | 123.61 | 28.54 | 583.28 | 2.016E-03 | 4.105E-07 | 186.602 | 1183.93 | 0.158 | 9.16E+05 | 9.72E-08 | | |
| 35 | 21.0 | 35.4 | 124.32 | 28.54 | 583.99 | 2.014E-03 | 4.109E-07 | 187.512 | 1184.64 | 0.158 | 9.19E+05 | 9.71E-08 | | |
| 36 | 0.2 | 35.0 | 127.68 | 28.54 | 587.35 | 2.002E-03 | 4.127E-07 | 186.998 | 1188.05 | 0.157 | 9.07E+05 | 9.65E-08 | | |
| 36 | 5.6 | 34.9 | 128.39 | 28.54 | 588.06 | 1.999E-03 | 4.131E-07 | 186.843 | 1188.76 | 0.157 | 9.04E+05 | 9.64E-08 | | |

# Computed Test Conditions

| run | alpcor | qc | temp | baro | T-R | rho | mu | V | Vsound | Mach | Rn/ft | rho-ast | rho ave | Mave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 11.0 | 35.0 | 128.39 | 28.54 | 588.06 | 2.000E-03 | 4.131E-07 | 187.097 | 1188.76 | 0.157 | 9.06E+05 | 9.64E-08 | 9.64E-08 | 0.158 |
| 36 | 16.0 | 35.2 | 128.74 | 28.54 | 588.41 | 1.998E-03 | 4.133E-07 | 187.701 | 1189.12 | 0.158 | 9.08E+05 | 9.64E-08 | | |
| 36 | 21.0 | 35.5 | 129.10 | 28.54 | 588.77 | 1.997E-03 | 4.134E-07 | 188.555 | 1189.48 | 0.159 | 9.11E+05 | 9.63E-08 | | |
| 37 | 0.4 | 35.1 | 114.77 | 28.54 | 574.44 | 2.047E-03 | 4.057E-07 | 185.195 | 1174.92 | 0.158 | 9.34E+05 | 9.87E-08 | | |
| 37 | 11.0 | 34.9 | 120.79 | 28.55 | 580.46 | 2.026E-03 | 4.089E-07 | 185.592 | 1181.05 | 0.157 | 9.20E+05 | 9.77E-08 | | |
| 37 | 16.1 | 35.3 | 123.08 | 28.54 | 582.75 | 2.018E-03 | 4.102E-07 | 187.048 | 1183.39 | 0.158 | 9.20E+05 | 9.73E-08 | 9.75E-08 | 0.158 |
| 37 | 21.1 | 35.6 | 125.03 | 28.54 | 584.70 | 2.011E-03 | 4.112E-07 | 188.154 | 1185.36 | 0.159 | 9.20E+05 | 9.70E-08 | | |
| 37 | 26.0 | 35.7 | 126.80 | 28.53 | 586.47 | 2.005E-03 | 4.122E-07 | 188.729 | 1187.15 | 0.159 | 9.18E+05 | 9.67E-08 | | |
| 38 | 0.4 | 34.9 | 128.39 | 28.53 | 588.06 | 1.999E-03 | 4.131E-07 | 186.856 | 1188.76 | 0.157 | 9.04E+05 | 9.64E-08 | | |
| 38 | 6.0 | 35.2 | 128.57 | 28.54 | 588.24 | 1.999E-03 | 4.132E-07 | 187.672 | 1188.94 | 0.158 | 9.08E+05 | 9.64E-08 | | |
| 38 | 11.1 | 35.2 | 129.10 | 28.54 | 588.77 | 1.997E-03 | 4.134E-07 | 187.757 | 1189.48 | 0.158 | 9.07E+05 | 9.63E-08 | 9.62E-08 | 0.158 |
| 38 | 16.2 | 35.2 | 130.16 | 28.53 | 589.83 | 1.993E-03 | 4.140E-07 | 187.952 | 1190.55 | 0.158 | 9.05E+05 | 9.61E-08 | | |
| 38 | 21.1 | 35.6 | 131.40 | 28.53 | 591.07 | 1.989E-03 | 4.147E-07 | 189.215 | 1191.80 | 0.159 | 9.07E+05 | 9.59E-08 | | |
| 39 | 0.4 | 35.0 | 131.93 | 28.52 | 591.60 | 1.987E-03 | 4.150E-07 | 187.711 | 1192.33 | 0.157 | 8.99E+05 | 9.58E-08 | | |
| 39 | 5.9 | 35.3 | 132.28 | 28.53 | 591.95 | 1.986E-03 | 4.152E-07 | 188.557 | 1192.69 | 0.158 | 9.02E+05 | 9.58E-08 | | |
| 39 | 11.0 | 35.2 | 132.81 | 28.52 | 592.48 | 1.984E-03 | 4.154E-07 | 188.387 | 1193.22 | 0.158 | 9.00E+05 | 9.57E-08 | 9.57E-08 | 0.158 |
| 39 | 16.1 | 35.3 | 132.81 | 28.53 | 592.48 | 1.984E-03 | 4.154E-07 | 188.642 | 1193.22 | 0.158 | 9.01E+05 | 9.57E-08 | | |
| 39 | 21.1 | 35.5 | 133.87 | 28.52 | 593.54 | 1.980E-03 | 4.160E-07 | 189.371 | 1194.29 | 0.159 | 9.01E+05 | 9.55E-08 | | |
| 40 | 0.4 | 34.5 | 123.08 | 28.52 | 582.75 | 2.016E-03 | 4.102E-07 | 184.993 | 1183.39 | 0.156 | 9.09E+05 | 9.72E-08 | | |
| 40 | 11.1 | 34.8 | 128.21 | 28.52 | 587.88 | 1.999E-03 | 4.130E-07 | 186.585 | 1188.58 | 0.157 | 9.03E+05 | 9.64E-08 | | |
| 40 | 16.1 | 35.1 | 129.98 | 28.52 | 589.65 | 1.993E-03 | 4.139E-07 | 187.683 | 1190.37 | 0.158 | 9.04E+05 | 9.61E-08 | 9.62E-08 | 0.158 |
| 40 | 21.1 | 35.4 | 131.75 | 28.52 | 591.42 | 1.987E-03 | 4.149E-07 | 188.765 | 1192.15 | 0.158 | 9.04E+05 | 9.58E-08 | | |
| 40 | 26.1 | 35.8 | 133.16 | 28.52 | 592.83 | 1.982E-03 | 4.156E-07 | 190.043 | 1193.58 | 0.159 | 9.06E+05 | 9.56E-08 | | |
| 41 | 0.4 | 34.8 | 134.40 | 28.52 | 594.07 | 1.978E-03 | 4.163E-07 | 187.565 | 1194.82 | 0.157 | 8.91E+05 | 9.54E-08 | | |
| 41 | 6.0 | 34.8 | 134.93 | 28.52 | 594.60 | 1.977E-03 | 4.166E-07 | 187.649 | 1195.36 | 0.157 | 8.90E+05 | 9.53E-08 | | |
| 41 | 11.1 | 34.9 | 134.93 | 28.52 | 594.60 | 1.977E-03 | 4.166E-07 | 187.918 | 1195.36 | 0.157 | 8.92E+05 | 9.53E-08 | 9.53E-08 | 0.157 |
| 41 | 16.2 | 35.1 | 135.11 | 28.52 | 594.78 | 1.976E-03 | 4.167E-07 | 188.484 | 1195.54 | 0.158 | 8.94E+05 | 9.53E-08 | | |
| 41 | 21.0 | 35.4 | 136.52 | 28.52 | 596.19 | 1.971E-03 | 4.174E-07 | 189.539 | 1196.96 | 0.158 | 8.95E+05 | 9.50E-08 | | |
| 42 | 0.4 | 35.3 | 136.88 | 28.53 | 596.55 | 1.971E-03 | 4.176E-07 | 189.275 | 1197.31 | 0.158 | 8.93E+05 | 9.50E-08 | | |
| 42 | 5.9 | 35.1 | 137.05 | 28.54 | 596.72 | 1.971E-03 | 4.177E-07 | 188.740 | 1197.49 | 0.158 | 8.90E+05 | 9.50E-08 | | |
| 42 | 11.0 | 35.3 | 137.41 | 28.53 | 597.08 | 1.969E-03 | 4.179E-07 | 189.359 | 1197.84 | 0.158 | 8.92E+05 | 9.50E-08 | 9.50E-08 | 0.158 |
| 42 | 16.2 | 35.3 | 137.41 | 28.53 | 597.08 | 1.969E-03 | 4.179E-07 | 189.359 | 1197.84 | 0.158 | 8.92E+05 | 9.50E-08 | | |
| 42 | 21.1 | 35.3 | 138.29 | 28.53 | 597.96 | 1.966E-03 | 4.184E-07 | 189.512 | 1198.73 | 0.158 | 8.90E+05 | 9.48E-08 | | |
| 43 | 0.3 | 35.0 | 91.43 | 28.62 | 551.10 | 2.140E-03 | 3.928E-07 | 180.873 | 1150.80 | 0.157 | 9.85E+05 | 1.03E-07 | | |
| 43 | 11.1 | 35.0 | 99.39 | 28.61 | 559.06 | 2.109E-03 | 3.972E-07 | 182.187 | 1159.08 | 0.157 | 9.67E+05 | 1.02E-07 | | |
| 43 | 16.1 | 35.8 | 102.22 | 28.62 | 561.89 | 2.099E-03 | 3.988E-07 | 184.697 | 1162.01 | 0.159 | 9.72E+05 | 1.01E-07 | 1.01E-07 | 0.158 |
| 43 | 21.1 | 35.5 | 104.69 | 28.61 | 564.36 | 2.089E-03 | 4.001E-07 | 184.352 | 1164.56 | 0.158 | 9.62E+05 | 1.01E-07 | | |
| 43 | 26.1 | 35.7 | 107.17 | 28.61 | 566.84 | 2.080E-03 | 4.015E-07 | 185.276 | 1167.12 | 0.159 | 9.60E+05 | 1.00E-07 | | |
| 44 | 0.4 | 34.9 | 109.11 | 28.62 | 568.78 | 2.073E-03 | 4.026E-07 | 183.477 | 1169.12 | 0.157 | 9.45E+05 | 1.00E-07 | | |
| 44 | 5.8 | 34.9 | 109.64 | 28.62 | 569.31 | 2.072E-03 | 4.029E-07 | 183.562 | 1169.66 | 0.157 | 9.44E+05 | 9.99E-08 | | |
| 44 | 11.0 | 35.0 | 110.17 | 28.63 | 569.84 | 2.070E-03 | 4.032E-07 | 183.885 | 1170.21 | 0.157 | 9.44E+05 | 9.98E-08 | 9.98E-08 | 0.157 |
| 44 | 16.1 | 35.3 | 111.24 | 28.63 | 570.91 | 2.066E-03 | 4.037E-07 | 184.844 | 1171.30 | 0.158 | 9.46E+05 | 9.96E-08 | | |
| 44 | 21.0 | 35.6 | 112.65 | 28.63 | 572.32 | 2.061E-03 | 4.045E-07 | 185.845 | 1172.75 | 0.158 | 9.47E+05 | 9.94E-08 | | |
| 45 | 0.4 | 35.0 | 114.24 | 28.63 | 573.91 | 2.056E-03 | 4.054E-07 | 184.528 | 1174.38 | 0.157 | 9.36E+05 | 9.91E-08 | | |
| 45 | 5.8 | 34.9 | 114.24 | 28.64 | 573.91 | 2.056E-03 | 4.054E-07 | 184.251 | 1174.38 | 0.157 | 9.35E+05 | 9.92E-08 | | |
| 45 | 10.9 | 35.1 | 114.95 | 28.64 | 574.62 | 2.054E-03 | 4.058E-07 | 184.892 | 1175.10 | 0.157 | 9.36E+05 | 9.90E-08 | 9.90E-08 | 0.158 |
| 45 | 16.1 | 35.2 | 115.30 | 28.64 | 574.97 | 2.052E-03 | 4.060E-07 | 185.213 | 1175.46 | 0.158 | 9.36E+05 | 9.90E-08 | | |

## APPENDIX B

### Fortran Programs

The following Fortran Programs are included:

| | |
|---|---|
| Reduce.for | Program written by NIAR to convert the voltage data to psi. |
| Filter.for | Program which applies a low-pass filter designed using a program from Parks, T. W., and Burrus, C. S., <u>Digital Filter Design</u>, John Wiley & Sons, Inc., New York, 1987 to the data. |
| Acftf.for | Program written to compute the frequency transfer functions of the data acquisition system using autocorrelations. |
| Autocor.for | Program which computes the power spectral densities of the buffet pressure using autocorrelations and takes account for the frequency transfer functions of the data acquisition system. |
| Mpsd.for | Program which computes the structural response. |

```
C
C                         MARK SMAGLIK
C        THE WALTER H. BEECH MEMORIAL 7X10 FOOT WIND TUNNEL
C                   WICHITA STATE UNIVERSITY
C                    DATA REDUCTION ROUTINE
C                          REDUCE.FOR
C                           10/10/94
C
C                   LAST REVISED:  10/19/94
C
C
C      THIS PROGRAM READS IN A SET OF DYNAMIC DATA, STORES THIS DATA IN
C      ARRAYS AND REDUCES THEM, USING KULITE SENSITIVITIES (VOLTS TO PSI).
C
C      VARIABLES:
C
C              CHAN0:             VARIABLE CONTAINING CHANNEL 0
C              CHAN1:             VARIABLE CONTAINING CHANNEL 1
C              CHAN2:             VARIABLE CONTAINING CHANNEL 2
C              CHAN5:             VARIABLE CONTAINING CHANNEL 5
C              CHAN6:             VARIABLE CONTAINING CHANNEL 6
C              CHAN8:             VARIABLE CONTAINING CHANNEL 8
C              CHAN9:             VARIABLE CONTAINING CHANNEL 9
C              CHAN10:            VARIABLE CONTAINING CHANNEL 10
C              CHAN11:            VARIABLE CONTAINING CHANNEL 11
C              CONVERSION(I):   ARRAY CONTAINING THE KULITE SENSITIVITIES
C              WOZBAL(I):       ARRAY CONTAINING THE WIND-OFF-ZERO DATA
C
C
C      FILES USED IN THIS PROGRAM:
C
C       WOZFILE
C
C           THIS FILE CONTAINS THE WIND OFF VALUES FOR THE DYNAMIC CHANNELS
C           AND IS WRITTEN IN THE FOLLOWING FORMAT.
C
C               RUN, CHAN0, CHAN1, CHAN2, CHAN5, CHAN6, ...  CHAN11
C
C      INFILE
C
C           THIS FILE CONTAINS THE DYNAMIC DATA (IN VOLTS) FOR
C           EACH DATA POINT AND IS WRITTEN IN THE FOLLOWING FORMAT:
C
C           CHAN0(1), CHAN1(1), CHAN2(1), CHAN5(1), CHAN6(1), ... CHAN11(1)
C           CHAN0(2), CHAN1(2), CHAN2(2), CHAN5(2), CHAN6(2), ... CHAN11(2)
C               .                          .                         .
C               .                          .                         .
C              ETC                        ETC                       ETC
C               .                          .                         .
C               .                          .                         .
C           CHAN0(8192),..................ETC.................CHAN11(8192)
C
C
C      OUTFILE
C
C           THIS FILE CONTAINS THE PROGRAM OUTPUT (REDUCED DATA) IN CSV
C           (COMMA-SEPARATED-VARIABLE) FORMAT AND IS WRITTEN AS THE
C           FOLLOWING:
C
```

```
C              CHAN0(1), CHAN1(1), CHAN2(1), CHAN5(1), CHAN6(1),   ... CHAN11(1)
C              CHAN0(2), CHAN1(2), CHAN2(2), CHAN5(2), CHAN6(2),   ... CHAN11(2)
C                  .                            .                          .
C                  .                            .                          .
C                 ETC                          ETC                        ETC
C                  .                            .                          .
C                  .                            .                          .
C
CHAN0(8192),..................ETC..................CHAN11(8192)
C
C              NOTE:  CHAN0,THROUGH CHAN11 HAVE THE UNITS OF (PSI)
C
C     ******************************************************************
C     *                                                                *
C     *                         MAIN PROGRAM                           *
C     *                                                                *
C     ******************************************************************
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 CONVERSION(9), WOZBAL(9), RUNNUM, NUMBER
      INTEGER I
      CHARACTER*12 INFILE,OUTFILE,WOZFILE
      LOGICAL ERROR
C
C
C  INITIALIZE KULITE SENSITIVITIES
C
      DATA (CONVERSION(I),I=1,9) /9.241E-3,9.389E-3,9.188E-3,9.123E-3,
     $         9.471E-3, 8.78E-3,9.128E-3,8.962E-2,9.186E-3/
C
C  INITIALIZE WIND-OFF-ZERO ARRAY COMES FROM A SEPARATE DATA FILE
C
      LASTRUN=999
      WOZFILE="DATAWOZ.CSV"
      OPEN (UNIT=5,FILE='e:\datawoz.csv',STATUS='OLD')
c      OPEN (UNIT=5,FILE=WOZFILE,STATUS='OLD')
      PRINT*, 'ENTER THE RUN NUMBER TO BE PROCESSED OR 999 TO QUIT: '
      READ(*,*) NUMBER
C
      IF (NUMBER.EQ.999) GOTO 99
C
      DO 1 J=1, LASTRUN
        READ (5,*) RUNNUM,(WOZBAL(K),K=1,9)
    1   IF (RUNNUM.EQ.NUMBER) GOTO 2
C
C
    2 ERROR = .FALSE.
      N = 8192
C
C  THE FOLLOWING LINES OF CODE PROMPT THE USER FOR THE INPUT FILE NAME,
C  APPEND THE MS-DOS FILE EXTENSION '.RED' TO SAID FILE NAME &
C  USE THE NEW FILE NAME AS THE OUTPUT FILE
C
    5 PRINT*, 'ENTER THE INPUT FILE NAME (DYNxxPx.CSV) OR Q TO QUIT: '
      READ(*,3) INFILE
    3 FORMAT (A12)
C
      IF ((INFILE.EQ.'Q').or.infile.eq.'q') GOTO 99
```

```
C
      DO 7 I=1,LEN(INFILE)
        IF(INFILE(I:I).EQ.'.') GOTO 8
    7 CONTINUE
C
    8 OUTFILE = INFILE(:I-1) // '.RED'
      WRITE(*,11) INFILE, OUTFILE
   11 FORMAT(//,' READING FROM: ',A12,3X,'WRITING TO: ',A12)
C
      OPEN (UNIT=1,FILE=INFILE,STATUS='OLD')
      OPEN (UNIT=9,FILE=OUTFILE,STATUS='NEW')
C
C  READ DATA FROM DATA FILE
C
      DO 10 I=1,N
C
       READ(1,*) CHAN0,CHAN1,CHAN2,CHAN5,CHAN6,CHAN8,CHAN9,CHAN10,CHAN11
C
C  SUBTRACT WIND-OFF-ZEROS AND CONVERT THE VOLTAGES TO PSI
C
        CHAN0 = ( CHAN0  - WOZBAL(1) )/CONVERSION(1)
        CHAN1 = ( CHAN1  - WOZBAL(2) )/CONVERSION(2)
        CHAN2 = ( CHAN2  - WOZBAL(3) )/CONVERSION(3)
        CHAN5 = ( CHAN5  - WOZBAL(4) )/CONVERSION(4)
        CHAN6 = ( CHAN6  - WOZBAL(5) )/CONVERSION(5)
        CHAN8 = ( CHAN8  - WOZBAL(6) )/CONVERSION(6)
        CHAN9 = ( CHAN9  - WOZBAL(7) )/CONVERSION(7)
        CHAN10= ( CHAN10 - WOZBAL(8) )/CONVERSION(8)
        CHAN11= ( CHAN11 - WOZBAL(9) )/CONVERSION(9)
C
C  PRINT THE DATA TO ITS RESPECTIVE FILE IN CSV (COMMA-SEPARATED-VARIABLE)
FORMAT
C
        WRITE(9,100) CHAN0,CHAN1,CHAN2,CHAN5,CHAN6,CHAN8,CHAN9,
     $          CHAN10,CHAN11
  100   FORMAT(F10.4,8(',',F10.4))
C
   10 CONTINUE
C
C  CLOSE INPUT AND OUTPUT FILES
C
      CLOSE(UNIT=9)
      CLOSE(UNIT=1)
C
      GOTO 5
C
   99 IF ( ERROR ) THEN
        PRINT*, 'REDUCE.FOR:  ERROR READING DATA, SAMPLE ',I
      ELSE
        PRINT*, 'REDUCE.FOR:  SUCCESSFUL TERMINATION BY USER'
      ENDIF
C
  999 STOP
      END
C
C     *********************  END OF REDUCE.FOR  **********************
```

```
c
c
c
program filter2
      implicit real*8 (a-h,o-z)
      character*12 infile,outfile
c
c     direct convolution FIR filter
c
      dimension h(380),t(8192,9),temp(8192)
      m=380
c

c
c     coef.out -- 101,0.4,0.5,1
c     coef2.out --- 101,0.35,0.4,1
c     coef3.out --- 301,0.35,0.4,1
c     coef4.out --- 301,0.35,0.4,1
c     coef5.out --- 380,0.35,0.4,1
c     h1.out -------301 1 4
c                 0.0 0.01 0.02 0.15 0.16 0.17 0.18 0.5
c         values  0. 1. 0. 1.
c         weights 10 1 10 1
c     low.out 380 .15 .16 1
c     hm.out 380 1 3
c            0.0 .16 .17 .35 .36 .5
c            0. 1. 0.
c            3. 1. 3.
c
      open(12,file='e:\low.out',status='old')
      read(12,*)(h(k),k=1,m/2)
      close(12)
      do 1 i=1,m/2
          h(m-i+1)=h(i)
    1 continue


c     open data file infile
c     output file outfile
c
    2 write(*,*)'  The name of data file to be filtered, or q to quit'
      read(*,500)infile
  500 format(a12)
      if(infile.eq.'q'.or.infile.eq.'Q')goto 99
      do 7 i=1,len(infile)
          if(infile(i:i).eq.'.')goto 8
    7 continue
    8 outfile=infile(:i-1)//'.flo'
      write(*,11)infile,outfile
   11 format(//, 'filtering:  'a12,3x,'writing new data to:  ',a12)

      open(8,file=infile,status='old')
      open(9,file=outfile,status='unknown')
c      do 31 j2=1,3
      do 20 j=1,8192
          read(8,*)  (t(j,ii),ii=1,9)
   20 continue
c      close(8)
      do 30 ii=1,9
```

```
          do 25 j=1,8192
               temp(j)=t(j,ii)
 25         continue
            call fltrm(temp,8192,h,m)
            do 27 j=1,8192
               t(j,ii)=temp(j)
 27         continue
 30   continue
c
c     done with low-pass filter
c
      do 40 j=1,8192
          write(9,100) (t(j,ii),ii=1,9)
 40   continue
 31   continue
      close(9)
      goto 2
 99   continue
100   format(1x,9(f14.8,1x))
      stop
      end

      subroutine fltrm(sig,np,a,n)
      implicit real*8 (a-h,o-z)
      dimension sig(1),a(1)
c      write(*,*)'np= ',np,'n= ',n
      nnp=np-n+1
      do 10 i=1,nnp
      ki=i-1
      y=0.0
      do 20 j=1,n
      y=y+a(j)*sig(ki+j)
 20   continue
      sig(i)=y
 10   continue
      nnp=nnp+1
      do 30 i=nnp,np
      sig(i)=0.
 30   continue
      return
      end
```

```
c
c     November 18, 1994 12:05AM
c
c     acftf2.for
c     this code computes the average autocorrelation function
c     over nd segments
c     then gets the power spectral density by FFT
c     then computes the FTF from that
c
c



C     analysis code for time series data
C     np is the number of runs
C     nd is number of segments
C     n is number of points per run
C
      program acftf
c      implicit real*8 (a-h,o-z)
      parameter (n=512,np=6,nd=8,irmax=511)
c     t matrix is temp which store time histories
      dimension r(np,nd,irmax+1),x(np,nd,n),
     +          xi(nd,n),
     +          aver(np,irmax+1),rap(nd*2)
      real*8 ro(2*(irmax+1)),ri(2*(irmax+1)),
     +       roc(2*(irmax+1)),ric(2*(irmax+1))
      complex*16 fft1(2*(irmax+1)),fft2(2*(irmax+1))
      character*12 flnm,flnmi,flnmh1
c
c
    2 continue
      write(*,*)' The name of data file containing output, or q to quit'
      read(*,500)flnm
  500 format(a12)
      if(flnm.eq.'q'.or.flnm.eq.'Q')goto 99
      write(*,*)' The name of file containing white noise'
      read(*,500)flnmi
      do 7 i=1,len(flnm)
          if(flnm(i:i).eq.'.')goto 8
    7 continue
    8 continue
      flnmh1=flnm(:i-1)//'.hc3'
      write(*,501)flnm,flnmi
      write(*,502)
      write(*,503)flnmh1
  501 format(//, 3x,'   reading:   ',3x,a12,3x,a12)
  502 format(3x,'   writing to:   ',a12)
  503 format(18x,a12)
c
      open (9,file=flnm , status='old')
      open(10,file=flnmi, status='old')
      open(22,file=flnmh1,status='unknown')
c
c     read in all time histories
c
      do 10 j2=1,nd
      do 10 j=1,n
          read(9,*) (x(ii,j2,j),ii=1,np)
```

```
          read(9,*)  (rap(ii),ii=1,np)
   10 continue
      close(9)
      do 11 j=1,n
          read(10,*)(xi(ii,j),ii=1,nd)
          read(10,*)(rap(ii),ii=1,nd)
   11 continue
      close(10)
c
c     initialize arrays
c
      do 50 ir=0,irmax
          ri(ir+1)=0.0
      do 50 j=1,np
          aver(j,ir+1)=0.0
      do 50 i2=1,nd
          r(j,i2,ir+1)=0.0
   50 continue
      do 100 i2=1,nd
          do 80 j=1,np
              do 70 ir=0,irmax
                  do 60 k=1,n-ir
                    r(j,i2,ir+1)=r(j,i2,ir+1)+x(j,i2,k)*x(j,i2,k+ir)
   60             continue
                  r(j,i2,ir+1)=r(j,i2,ir+1)/real(n)
                  aver(j,ir+1)=aver(j,ir+1)+r(j,i2,ir+1)
   70         continue
   80     continue
  100 continue
      reand=1.0/real(nd)
      do 110 j=1,np
      do 110 ir=0,irmax
      aver(j,ir+1)=aver(j,ir+1)*reand
  110 continue
      reanp=1.0/real(np)
      do 125 ir=0,irmax
          do 120 j=1,np
              ro(ir+1)=ro(ir+1)+aver(j,ir+1)
  120     continue
          ro(ir+1)=ro(ir+1)*reanp
  125 continue

      do 130 j=1,nd
             do 130 ir=0,irmax
                 do 130 k=1,n-ir
                     ri(ir+1)=ri(ir+1)+xi(j,k)*xi(j,k+ir)
  130 continue
      do 140 ir=0,irmax
          ri(ir+1)=ri(ir+1)/(real(n*nd))
  140 continue
c
      do 150 ir=0,irmax
           ric(ir+irmax+2)=ri(ir+1)
           ric(ir+2)=ri(irmax-ir+1)
           roc(ir+irmax+2)=ro(ir+1)
           roc(ir+2)=ro(irmax-ir+1)
  150 continue
           ric(1)=0.0
           roc(1)=0.0
```

```
      write(*,*)'done with autocorrellation'
      call twofft(ric,roc,fft1,fft2,2*(irmax+1))
      write(*,*)'done ffts - computing h and writing'
      do 300 ir=0,2*(irmax+1)
          write(22,504)real(ir)/(4.*8.26e-04*real(irmax+1)),
     +                     fft2(ir+1)/fft1(ir+1)
300   continue
504   format(1x,20(E13.6,1x))
      close(22)
      go to 2
 99   continue
      stop
      end
```

```
c
c     November 22, 1994 4:00PM
c
c     autocor9..for
c     this code computes the average autocorrelation function
c     over nd segments
c     then gets the power spectral density by FFT
c
c     analysis code for time series data
c     np is the number of runs
c     nd is number of segments
c     n is number of points per run
c
      program bstats
c      implicit real*8 (a-h,o-z)
      parameter (n=512,np=9,nd=8,irmax=511)
c     t matrix is temp which store time histories
c     ft is fft of t matrix (complex)
c     rmean is the mean of each transducer for each run
c     var is the variance of each trans. for each run
c
      dimension r(np,np,nd,irmax+1),x(np,nd,n),temp1(2*(irmax+1)),
     +          temp2(2*(irmax+1)),gam2(np,np,2*(irmax+1)),
     +          aver(np,np,irmax+1),h(np,2*(irmax+1)),rap(np),
     +          rmean(np,nd),rms(np,nd),var(np,nd)
      complex fft1(2*(irmax+1)),fft2(2*(irmax+1)),ft(np,np,2*(irmax+1))
      character*12 flnm,
     +            flnmg1,flnmg2,flnmg3,
     +            flnmg4,flnmg5,flnmg6,
     +            flnmg7,flnmg8,flnmm,
     +            flnms1,flnms2,flnms3,
     +            flnms4,flnms5,flnms6,
     +            flnms7,flnms8,flnms9
c
c
    2 continue
      write(*,*)'The name of data file for processing, or q to quit'
      read(*,500)flnm
  500 format(a12)
      if(flnm.eq.'q'.or.flnm.eq.'Q')goto 99
      do 7 i=1,len(flnm)
          if(flnm(i:i).eq.'.')goto 8
    7 continue
    8 continue
      flnms1=flnm(:i-1)//'.s0'
      flnms2=flnm(:i-1)//'.s1'
      flnms3=flnm(:i-1)//'.s2'
      flnms4=flnm(:i-1)//'.s5'
      flnms5=flnm(:i-1)//'.s6'
      flnms6=flnm(:i-1)//'.s8'
      flnms7=flnm(:i-1)//'.s9'
      flnms8=flnm(:i-1)//'.s10'
      flnms9=flnm(:i-1)//'.s11'
      flnmg1=flnm(:i-1)//'.g0'
      flnmg2=flnm(:i-1)//'.g1'
      flnmg3=flnm(:i-1)//'.g2'
      flnmg4=flnm(:i-1)//'.g5'
      flnmg5=flnm(:i-1)//'.g6'
      flnmg6=flnm(:i-1)//'.g8'
```

```
      flnmg7=flnm(:i-1)//'.g9'
      flnmg8=flnm(:i-1)//'.g10'
      flnmm=flnm(:i-1)//'.mrv'
      write(*,501)flnm
      write(*,502)flnmg1
      write(*,503)flnmg2
      write(*,503)flnmg3
      write(*,503)flnmg4
      write(*,503)flnmg5
      write(*,503)flnmg6
      write(*,503)flnmg7
      write(*,503)flnmg8
  501 format(//, 3x,'  reading:  ',3x,a12)
  502 format(3x,'  writing to:  ',a12)
  503 format(18x,a12)
c
      open(9,file=flnm,status='old')
      open(10,file='d:\n94ftf\ftfall3.txt',status='old')
      open(22,file=flnms1,status='unknown')
      open(23,file=flnms2,status='unknown')
      open(24,file=flnms3,status='unknown')
      open(25,file=flnms4,status='unknown')
      open(26,file=flnms5,status='unknown')
      open(27,file=flnms6,status='unknown')
      open(28,file=flnms7,status='unknown')
      open(29,file=flnms8,status='unknown')
      open(30,file=flnms9,status='unknown')
      open(31,file=flnmm,status='unknown')
      open(32,file=flnmg1,status='unknown')
      open(33,file=flnmg2,status='unknown')
      open(34,file=flnmg3,status='unknown')
      open(35,file=flnmg4,status='unknown')
      open(36,file=flnmg5,status='unknown')
      open(37,file=flnmg6,status='unknown')
      open(38,file=flnmg7,status='unknown')
      open(39,file=flnmg8,status='unknown')
c
c     read in all time histories
c
      do 10 j2=1,nd
      do 10 j=1,n
          read(9,*) (x(ii,j2,j),ii=1,np)
          read(9,*) (rap(ii),ii=1,np)
   10 continue
      close(9)
      do 11 j=1,2*(irmax+1)
          read(10,*)(h(ii,j),ii=1,np)
   11 continue
      close(10)
      do 20 j=1,2*(irmax+1)
      do 20 ii=1,np
          h(ii,j)=sqrt(h(ii,j))
   20 continue
c
c     initialize rmean,rms, and var
c
      do 22 j2=1,nd
      do 22 ii=1,np
          rmean(ii,j2)=0.0
```

```
         rms(ii,j2)=0.0
         var(ii,j2)=0.0
   22 continue
c
c     compute the mean and root mean square of raw data
c
      do 26 j2=1,nd
      do 26 ii=1,np
      do 25 j=1,n
         rmean(ii,j2)=rmean(ii,j2)+x(ii,j2,j)
         rms(ii,j2)=rms(ii,j2)+x(ii,j2,j)*x(ii,j2,j)
   25 continue
         rmean(ii,j2)=rmean(ii,j2)/real(n)
         rms(ii,j2)=sqrt(rms(ii,j2))/real(n)
   26 continue
c
c     compute variance
c
      do 31 j2=1,nd
      do 31 ii=1,np
      do 30 j=1,n
         var(ii,j2)=x(ii,j2,j)**2
   30 continue
         var(ii,j2)=var(ii,j2)/real(n-1)
   31 continue
      write(31,*)'mean    rms     var     column     segment'
      do 32 ii=1,np
      do 32 j2=1,nd
         write(31,*)rmean(ii,j2),rms(ii,j2),var(ii,j2),ii,j2
   32 continue
c
c     initialize autocorrelation variables
c
      do 50 i2=1,nd
      do 50 i=1,np
      do 50 ir=0,irmax
         r(i,i,i2,ir+1)=0.0
         aver(i,i,ir+1)=0.0
   50 continue
c
c     compute autocorrelations
c
      do 100 i2=1,nd
      do 90 i=1,np
         do 80 j=1,np
            do 70 ir=0,irmax
               do 60 k=1,n-ir
                 r(i,j,i2,ir+1)=r(i,j,i2,ir+1)+x(i,i2,k)*x(j,i2,k+ir)
   60          continue
               r(i,j,i2,ir+1)=r(i,j,i2,ir+1)/(real(n))
               aver(i,j,ir+1)=aver(i,j,ir+1)+r(i,j,i2,ir+1)
   70       continue
   80    continue
   90 continue
  100 continue
      reand=1.0/real(nd)
      do 110 i=1,np
      do 110 j=1,np
      do 110 ir=0,irmax
```

```
      aver(i,j,ir+1)=aver(i,j,ir+1)*reand
  110 continue
c       do 120 i=1,np
c       do 120 j=1,np
c       do 120 ir=0,irmax
c       aver(i,j,ir+1)=aver(i,j,ir+1)/aver(i,j,1)
c  120 continue
       write(*,*)'done with autocorrellation'
c
c      shove it through a FFT
c
       do 200 i=1,np,2
           do 200 j=i,np
               do 180 ir=0,irmax
c
c      flipping R using symmetries to get + and - tau
c
                   if(i.eq.9)then
                     temp1(ir+irmax+2)=aver(i,j,ir+1)
                     temp1(ir+2)=aver(j,i,irmax-ir+1)
                     temp2(ir+1)=0.0
                     temp2(ir+irmax+1)=0.0
                   else
                     temp1(ir+irmax+2)=aver(i,j,ir+1)
                     temp1(ir+2)=aver(j,i,irmax-ir+1)
                     temp2(ir+irmax+2)=aver(i+1,j,ir+1)
                     temp2(ir+2)=aver(j,i+1,irmax-ir+1)
                   endif
  180          continue
           temp1(1)=0.0
           temp2(1)=0.0
           call twofft(temp1,temp2,fft1,fft2,2*(irmax+1))
           do 190 ir=1,2*(irmax+1)
               if(i.eq.9)then
c                   ft(i,j,ir)=fft1(ir)/(h(i,ir)*h(j,ir))
                   ft(i,j,ir)=fft1(ir)
               else
c                   ft(i,j,ir)=  fft1(ir)/(h(i,ir)*h(j,ir))
c                   ft(i+1,j,ir)=fft2(ir)/(h(i+1,ir)*h(j,ir))
                    ft(i,j,ir)=  fft1(ir)
                    ft(i+1,j,ir)=fft2(ir)
               endif
  190      continue
  200 continue
       write(*,*)'done ffts - writing files'
       do 300 ir=0,irmax+1,2
           do 300 i=1,np
            write(22+i-1,504)real(ir)/(4.*8.26e-04*real(irmax+1)),
     +             (ft(i,j,ir+1),j=i,np)
  300      continue
c
c      compute coherence
c
       do 400 ii=1,np
       do 400 i3=ii,np
       do 400 ir=0,irmax+1,2
           gam2(ii,i3,ir+1)=
     +   real((cabs(ft(ii,i3,ir+1)))**2/(ft(ii,ii,ir+1)*ft(i3,i3,ir+1)))
  400 continue
```

```
      do 410 ir=0,irmax+1,2
      do 410 ii=1,np-1
           write(32+ii-1,504)real(ir)/(4.*8.26e-04*real(irmax+1)),
     +               (gam2(ii,i3,ir+1),i3=ii+1,np)
410   continue
504   format(1x,20(E13.6,1x))
      close(22)
      close(23)
      close(24)
      close(25)
      close(26)
      close(27)
      close(28)
      close(29)
      close(30)
      close(31)
      close(32)
      close(33)
      close(34)
      close(35)
      close(36)
      close(37)
      close(38)
      close(39)
      go to 2
 99   continue
      stop
      end
```

```
c
c
c     program to compute modal psd's
c         mpsd.for 12/23/94 11:00A
c
      program mpsd
      implicit real*8 (a-h,o-z)
      parameter(nm=3,nd=14,nmax=300)
      real*8 n,ntrans
      dimension out(nmax,nmax),b(1,4),n(nd,6),ntrans(6,nd),x(nd),y(nd),
     +          phi(nd,nm),phitran(nm,nd),t1(20),temp(nmax,nmax),
     +          rnm(1,nm),rny(1,nm),rnt(1,nm),
     +          rnmtran(nm,1),rnytran(nm,1),rnttran(nm,1),
     +          h1(4,4),h2(4,4)
      complex*16 sfm(6,6),sf(nmax,nmax),temp2(nmax,nmax),
     +           temp3(nmax,nmax),
     +           sn(nmax,nmax),hc(nmax,nmax),ht(nmax,nmax),
     +           sm(nmax,nmax),sy(nmax,nmax),st(nmax,nmax),sm0,sy0,st0
      character direc*8,runno*2,pt*2,flnm*21,side*1,
     +            flnm1*25,flnm2*25,flnm3*25,flnm4*25,
     +            flnm5*25,flnm6*25,flnmh*20
      data direc /'e:\dyn0'/
      data x /14.57143d0,20.14286d0,25.71429d0,31.28571d0,
     +        36.85714d0,42.42857d0,48.d0,39.57143d0,
     +        43.14286d0,46.71429d0,50.28571d0,53.85714d0,
     +        57.42857d0,61.0d0/
      data y /15.71429d0,31.42857d0,47.14286d0,62.85714d0,
     +        78.57143d0,94.28571d0,110.d0,15.71429d0,31.42857d0,
     +        47.14286d0,62.85714d0,78.57143d0,94.28571d0,110.d0/
      data rnm /-28787.4d0,93382.2d0,25237.2d0/          !root bending moment
      data rny /0.976894d0,0.875172d0,0.280773d0/        !mean tip deflection
      data rnt /-0.003555d0,-0.019204d0,0.110650d0/      !tip twist
c
c     from b200freq1.out
c
      data phi /  1.81608d-2,8.11011d-2,1.91912d-1,
     +            3.44854d-1,5.30554d-1,7.37511d-1,
     +            9.53788d-1,3.23590d-2,1.13669d-1,      !mode 1
     +            2.37736d-1,3.97954d-1,5.85583d-1,
     +            7.89695d-1,1.0d0,

     +            -1.07207d-1,-3.24668d-1,-5.32947d-1,
     +            -5.83280d-1,-3.7017d-1,1.09875d-1,
     +            7.50447d-1,-1.29270d-1,-3.39555d-1,    !mode 2
     +            -4.87749d-1,-4.46213d-1,-1.49412d-1,
     +            3.70931d-1,1.0d0,

     +            1.79035d-1,3.81946d-1,5.30204d-1,
     +            6.56162d-1,7.85368d-1,9.11074d-1,
     +            1.0d0,-2.19901d-1,-5.09157d-1,          !mode 3
     +            -7.43562d-1,-8.67304d-1,-8.45735d-1,
     +            -6.87461d-1,-4.38135d-1 /

      data h1 /  3.0544,-0.0599,-0.0463,0.00091,
     +           -1.3176,0.0599,0.01996,-0.00091,
     +           1.1213,-0.0320,-0.0401,0.001144,
     +           -1.8581,0.0320,-0.06636,-0.00114/

      data h2 /17.652174,-0.304348,-0.229249,0.0039526,
```

```
     +         -10.65217,0.3043478,0.1383399,-0.003953,
     +         11.142857,-0.285714,-0.168831,0.004329,
     +         -17.14286,0.2857143,0.2597403,-0.004329/
c
c     read in run # and form directory name
c
      write(*,*)' input run # '
      read(*,500)runno
      write(*,*)' input run # again (just for fun)'
      read(*,*)nrunno
      direc(7:7)=runno(1:1)
      if(runno(2:2).ne.' ')direc(8:8)=runno(2:2)
c
c     read in point and form input file name
c
      write(*,*)'  input point (alpha) #'
      read(*,500)pt
      write(*,*)'  use left or right side data? (L or R)'
      read(*,501)side
      do 3 i=1,len_trim(direc)
          flnm(i:i)=direc(i:i)
    3 continue
      flnm=flnm(:i-1)//'\dyn'
      flnm(len_trim(flnm)+1:len_trim(flnm)+1)=runno(1:1)
      if(runno(2:2).ne.' ')
     +     flnm(len_trim(flnm)+1:len_trim(flnm)+1)=runno(2:2)
      flnm(len_trim(flnm)+1:len_trim(flnm)+1)='p'
      flnm(len_trim(flnm)+1:len_trim(flnm)+1)=pt(1:1)
      if(pt(2:2).ne.' ')
     +    flnm(len_trim(flnm)+1:len_trim(flnm)+1)=pt(2:2)
  500 format (a2)
  501 format (a1)
  502 format (1x,20(E13.6,1x))
  503 format (a20)
c
c     need to add suffix and open files with psd's
c         to get all six using mirror points where bad transducers are
c
      if(side.eq.'L'.or.side.eq.'l') then
          flnm1=flnm(:len_trim(flnm))//'.s0'
          flnm2=flnm(:len_trim(flnm))//'.s1'
          flnm3=flnm(:len_trim(flnm))//'.s2'
          flnm4=flnm(:len_trim(flnm))//'.s11'
          flnm5=flnm(:len_trim(flnm))//'.s10'
          flnm6=flnm(:len_trim(flnm))//'.s5'
      endif
      if(side.eq.'R'.or.side.eq.'r') then
          flnm1=flnm(:len_trim(flnm))//'.s6'
          flnm2=flnm(:len_trim(flnm))//'.s1'
          flnm3=flnm(:len_trim(flnm))//'.s8'
          flnm4=flnm(:len_trim(flnm))//'.s11'
          flnm5=flnm(:len_trim(flnm))//'.s10'
          if(nrunno.lt.66)flnm6=flnm(:len_trim(flnm))//'.s9'
          if(nrunno.ge.66)flnm6=flnm(:len_trim(flnm))//'.s5'
      endif
      open(30,file=flnm1,status='old')
      open(31,file=flnm2,status='old')
      open(32,file=flnm3,status='old')
      open(33,file=flnm4,status='old')
```

```
      open(34,file=flnm5,status='old')
      open(35,file=flnm6,status='old')
      write(*,*)'enter the name of heta file (full path)'
      read(*,503)flnmh
      open(40,file=flnmh,status='old')
c
c     get rn transposes
c
      do 44 i=1,nm
          rnmtran(i,1)=rnm(1,i)
          rnytran(i,1)=rny(1,i)
          rnttran(i,1)=rnt(1,i)
   44 continue
c
c     compute N
c         need to loop through all x,y pairs
c
      do 100 i=1,nd
          b(1,1)=1.0d0
          b(1,2)=x(i)
          b(1,3)=y(i)
          b(1,4)=x(i)*y(i)
          if(y(i).lt.66.)then
              call matmul(b,1,4,h1,4,out)
              n(i,1)=out(1,1)
              n(i,2)=out(1,4)
              n(i,3)=0.0
              n(i,4)=0.0
              n(i,5)=out(1,3)
              n(i,6)=out(1,2)
          else
              call matmul(b,1,4,h2,4,out)
              n(i,1)=0.0
              n(i,2)=out(1,1)
              n(i,3)=out(1,4)
              n(i,4)=out(1,3)
              n(i,5)=out(1,2)
              n(i,6)=0.0
          endif
  100 continue
c
c     so have n have measured psd's have modal matrix
c
c     need to compute modal force psd at each frequency
c         sf = phitran*n*sfm*ntrans*phi
c
c             phi       = modal matrix
c             n         = shape function matrix computed above
c             phitran   = transpose of phi
c             ntrans    = transpose of n
c             sfm       = measured psd's
c
      do 130 i=1,nd
      do 130 j=1,6
          ntrans(j,i)=n(i,j)
  130 continue
      do 150 i=1,nm         !nm = number of modes
      do 150 j=1,nd         !nd = number of nodes used in structural model
          phitran(i,j)=phi(j,i)
```

```
  150 continue
c       call matmul(phitran,nm,nd,n,6,temp)
c
c     freqency loop to get sfm
c
      do 999 i=1,154            ! # of frequencies
c
c     read in measured power spectral density and scale to full scale
c
      if(side.eq.'L'.or.side.eq.'l')then
          read(30,502)freq,(t1(j),j=1,18)
          sfm(1,1)=cmplx(t1(1))*12.d0
          sfm(1,2)=cmplx(t1(3),t1(4))*12.d0
          sfm(1,3)=cmplx(t1(5),t1(6))*12.d0
          sfm(1,4)=cmplx(t1(17),t1(18))*12.d0
          sfm(1,5)=cmplx(t1(15),t1(16))*12.d0
          sfm(1,6)=cmplx(t1(7),t1(8))*12.d0
          sfm(2,1)=conjg(sfm(1,2))
          sfm(3,1)=conjg(sfm(1,3))
          sfm(4,1)=conjg(sfm(1,4))
          sfm(5,1)=conjg(sfm(1,5))
          sfm(6,1)=conjg(sfm(1,6))
          read(31,502)freq,(t1(j),j=1,16)
          sfm(2,2)=cmplx(t1(1))*12.d0
          sfm(2,3)=cmplx(t1(3),t1(4))*12.d0
          sfm(2,4)=cmplx(t1(15),t1(16))*12.d0
          sfm(2,5)=cmplx(t1(13),t1(14))*12.d0
          sfm(2,6)=cmplx(t1(5),t1(6))*12.d0
          sfm(3,2)=conjg(sfm(2,3))
          sfm(4,2)=conjg(sfm(2,4))
          sfm(5,2)=conjg(sfm(2,5))
          sfm(6,2)=conjg(sfm(2,6))
          read(32,502)freq,(t1(j),j=1,14)
          sfm(3,3)=cmplx(t1(1))*12.d0
          sfm(3,4)=cmplx(t1(13),t1(14))*12.d0
          sfm(3,5)=cmplx(t1(11),t1(12))*12.d0
          sfm(3,6)=cmplx(t1(3),t1(4))*12.d0
          sfm(4,3)=conjg(sfm(3,4))
          sfm(5,3)=conjg(sfm(3,5))
          sfm(6,3)=conjg(sfm(3,6))
          read(33,502)freq,(t1(j),j=1,2)
          sfm(4,4)=cmplx(t1(1))*12.d0
          read(34,502)freq,(t1(j),j=1,4)
          sfm(5,5)=cmplx(t1(1),t1(2))*12.d0
          sfm(5,4)=cmplx(t1(3),t1(4))*12.d0
          sfm(4,5)=conjg(sfm(5,4))
          read(35,502)freq,(t1(j),j=1,12)
          sfm(6,6)=cmplx(t1(1))*12.d0
          sfm(6,4)=cmplx(t1(11),t1(12))*12.d0
          sfm(6,5)=cmplx(t1(9),t1(10))*12.d0
          sfm(4,6)=conjg(sfm(6,4))
          sfm(5,6)=conjg(sfm(6,5))
      endif
      if(side.eq.'R'.or.side.eq.'r')then
          read(30,502)freq,(t1(j),j=1,10)
          sfm(1,1)=cmplx(t1(1))*12.d0
          sfm(1,3)=cmplx(t1(3),t1(4))*12.d0
          sfm(1,4)=cmplx(t1(9),t1(10))*12.d0
          sfm(1,5)=cmplx(t1(7),t1(8))*12.d0
```

```
      if(nrunno.lt.66)sfm(1,6)=cmplx(t1(5),t1(6))*12.d0
      sfm(3,1)=conjg(sfm(1,3))
      sfm(4,1)=conjg(sfm(1,4))
      sfm(5,1)=conjg(sfm(1,5))
      if(nrunno.lt.66)sfm(6,1)=conjg(sfm(1,6))
      read(31,502)freq,(t1(j),j=1,16)
      sfm(2,2)=cmplx(t1(1))*12.d0
      sfm(2,1)=cmplx(t1(7),t1(8))*12.d0
      sfm(2,3)=cmplx(t1(9),t1(10))*12.d0
      sfm(2,4)=cmplx(t1(15),t1(16))*12.d0
      sfm(2,5)=cmplx(t1(13),t1(14))*12.d0
      if(nrunno.lt.66)sfm(2,6)=cmplx(t1(11),t1(12))*12.d0
      if(nrunno.ge.66)sfm(2,6)=cmplx(t1(5),t1(6))*12.d0
      sfm(1,2)=conjg(sfm(2,1))
      sfm(3,2)=conjg(sfm(2,3))
      sfm(4,2)=conjg(sfm(2,4))
      sfm(5,2)=conjg(sfm(2,5))
      sfm(6,2)=conjg(sfm(2,6))
      read(32,502)freq,(t1(j),j=1,8)
      sfm(3,3)=cmplx(t1(1))*12.d0
      sfm(3,4)=cmplx(t1(7),t1(8))*12.d0
      sfm(3,5)=cmplx(t1(5),t1(6))*12.d0
      if(nrunno.lt.66)sfm(3,6)=cmplx(t1(3),t1(4))*12.d0
      sfm(4,3)=conjg(sfm(3,4))
      sfm(5,3)=conjg(sfm(3,5))
      if(nrunno.lt.66)sfm(6,3)=conjg(sfm(3,6))
      read(33,502)freq,(t1(j),j=1,2)
      sfm(4,4)=cmplx(t1(1))*12.d0
      read(34,502)freq,(t1(j),j=1,4)
      sfm(5,5)=cmplx(t1(1),t1(2))*12.d0
      sfm(5,4)=cmplx(t1(3),t1(4))*12.d0
      sfm(4,5)=conjg(sfm(5,4))
      if(nrunno.lt.66)then
          read(35,502)freq,(t1(j),j=1,6)
          sfm(6,6)=cmplx(t1(1))*12.d0
          sfm(6,5)=cmplx(t1(3),t1(4))*12.d0
          sfm(6,4)=cmplx(t1(5),t1(6))*12.d0
          sfm(5,6)=conjg(sfm(6,5))
          sfm(4,6)=conjg(sfm(6,4))
      else
          read(35,502)freq,(t1(j),j=1,12)
          sfm(6,6)=cmplx(t1(1))*12.d0
          sfm(6,1)=cmplx(t1(3),t1(4))*12.d0
          sfm(6,3)=cmplx(t1(5),t1(6))*12.d0
          sfm(6,4)=cmplx(t1(11),t1(12))*12.d0
          sfm(6,5)=cmplx(t1(9),t1(10))*12.d0
          sfm(1,6)=conjg(sfm(6,1))
          sfm(3,6)=conjg(sfm(6,3))
          sfm(4,6)=conjg(sfm(6,4))
          sfm(5,6)=conjg(sfm(6,5))
      endif
  endif
  write(99,505)freq,((sfm(jjj,jjjj),jjj=1,6),jjjj=1,6)
  call cmatmul1(n,nd,6,sfm,6,temp2)             !real - complex
c  call cmatmul1(temp,nm,6,sfm,6,temp2)            !real - complex
  call cmatmul2(temp2,nm,6,ntrans,nd,temp3)     !complex - real
  write(98,505)freq,((temp3(jjj,jjjj),jjj=1,3),jjjj=1,3)
  call cmatmul2(temp3,nm,nd,phi,nm,sf)          !complex - real
c
```

```
c     now have sf for a given frequency corresponding to i {f=(i-1)*1.18228}
c
c     read in H at the same frequency
c
      do 200 j=1,nm
          read(40,*)(t1(j2),j2=1,2*nm)
          do 200 j2=1,2*nm,2
          ht((j2+1)/2,j)=cmplx(t1(j2),t1(j2+1))
  200 continue
      do 210 j=1,nm
      do 210 j2=1,nm
          hc(j,j2)=conjg(ht(j2,j))
  210 continue
          write(111,*)freq,real(hc(1,1)),imag(hc(1,1))
          write(112,*)freq,real(hc(2,2)),imag(hc(2,2))
          write(113,*)freq,real(hc(3,3)),imag(hc(3,3))
      call cmatmul3(hc,nm,nm,sf,nm,temp3)
      call cmatmul3(temp3,nm,nm,ht,nm,sn)
c
c     have modal response PSD {sn}
c
      call cmatmul15(rnm,1,nm,sn,nm,temp2)
      call cmatmul2(temp2,1,nm,rnmtran,1,sm)
      call cmatmul15(rny,1,nm,sn,nm,temp2)
      call cmatmul2(temp2,1,nm,rnytran,1,sy)
      call cmatmul15(rnt,1,nm,sn,nm,temp2)
      call cmatmul2(temp2,1,nm,rnttran,1,st)
      if(i.eq.1)sm0=1.0d0/sm(1,1)
      if(i.eq.1)sy0=1.0d0/sy(1,1)
      if(i.eq.1)st0=1.0d0/st(1,1)
      write(50,505)freq,sm(1,1),sy(1,1),st(1,1)
  505 format(1x,100(e13.6,1x))
  999 continue
      stop
      end
      subroutine matmul(x,m,n,y,n2,out)
      implicit real*8 (a-h,o-z)
      parameter(nmax=300)
c
c     multiply x(m,n) and y(n,n2) matrices result is out(m,n2)
c
      dimension x(m,n),y(n,n2),out(nmax,nmax)
      do 10 i=1,nmax
      do 10 j=1,nmax
          out(i,j)=0.0d0
   10 continue
      do 60 i=1,m
          do 50 j=1,n2
              do 40 j2=1,n
                  out(i,j)=out(i,j)+x(i,j2)*y(j2,j)
   40         continue
   50     continue
   60 continue
      return
      end

      subroutine cmatmul1(x,m,n,y,n2,out)
      implicit real*8 (a-h,o-z)
      parameter(nmax=300)
```

```
c
c     multiply x(m,n) and y(n,n2) matrices result is out(m,n2)
c
c      complex y(n,n2),out(nmax,nmax)
      dimension x(nmax,nmax)
      complex*16 y(n,n2),out(nmax,nmax)
      do 10 i=1,nmax
      do 10 j=1,nmax
          out(i,j)=0.0d0
   10 continue
      do 60 i=1,m
          do 50 j=1,n2
              do 40 j2=1,n
                  out(i,j)=out(i,j)+x(i,j2)*y(j2,j)
   40         continue
   50     continue
   60 continue
      return
      end

      subroutine cmatmul15(x,m,n,y,n2,out)
      implicit real*8 (a-h,o-z)
      parameter(nmax=300)
c
c     multiply x(m,n) and y(n,n2) matrices result is out(m,n2)
c
c      complex y(n,n2),out(nmax,nmax)
      dimension x(m,n)
      complex*16 y(nmax,nmax),out(nmax,nmax)
      do 10 i=1,nmax
      do 10 j=1,nmax
          out(i,j)=0.0d0
   10 continue
      do 60 i=1,m
          do 50 j=1,n2
              do 40 j2=1,n
                  out(i,j)=out(i,j)+x(i,j2)*y(j2,j)
   40         continue
   50     continue
   60 continue
      return
      end

      subroutine cmatmul2(x,m,n,y,n2,out)
      implicit real*8 (a-h,o-z)
      parameter(nmax=300)
c
c     multiply x(m,n) and y(n,n2) matrices result is out(m,n2)
c
c      complex x(m,n),out(nmax,nmax)
      dimension y(n,n2)
      complex*16 x(nmax,nmax),out(nmax,nmax)
      do 10 i=1,nmax
      do 10 j=1,nmax
          out(i,j)=0.0d0
   10 continue
      do 60 i=1,m
          do 50 j=1,n2
              do 40 j2=1,n
```

```
                 out(i,j)=out(i,j)+x(i,j2)*y(j2,j)
 40          continue
 50      continue
 60  continue
     return
     end

     subroutine cmatmul3(x,m,n,y,n2,out)
     implicit real*8 (a-h,o-z)
     parameter(nmax=300)
c
c    multiply x(m,n) and y(n,n2) matrices result is out(m,n2)
c
c     complex x(m,n),y(n,n2),out(nmax,nmax)
     complex*16 x(nmax,nmax),y(nmax,nmax),out(nmax,nmax)
     do 10 i=1,nmax
     do 10 j=1,nmax
         out(i,j)=0.0
 10  continue
     do 60 i=1,m
         do 50 j=1,n2
             do 40 j2=1,n
                 out(i,j)=out(i,j)+x(i,j2)*y(j2,j)
 40          continue
 50      continue
 60  continue
     return
     end
```

APPENDIX C

Sample ASTROS Input Files

The ASTROS files that are included are:

| | |
|---|---|
| b200freq.inp | File sequence was used for modal analysis to determine frequencies and mode shapes. |
| b200heta134-1.inp | File sequence is modified gust analysis to determine the modal complex frequency response matrix [H($\omega$)]. This is an example file; similar files were used for different mach numbers. The current structure of ASTROS required the frequency range to be split, running half per run. |
| b200nsigma.inp | Modified gust analysis to determine the tip displacement, rotations, and root bending moment [N] matrices. |

```
ASSIGN DATABASE PROB3 RAC NEW DELETE
SOLUTION
  TITLE   =Beech King Air Horizontal Tail Model
  SUBTITLE=Frequencies and Mode Shapes
  PRINT ROOT(MODES=ALL)=ALL,DISP(MODES=ALL,TIME=ALL)=ALL,VELO(TIME=ALL)=ALL
  ANALYZE
     BOUNDARY METHOD=900, REDUCE=3, SPC=1
        STATICS ( MECH=1 )
          LABEL=STATIC ANALYSIS FOR UNIT MOMENT FOR PART 1
        MODES
          LABEL=MODAL ANALYSIS FOR PART 1
END
BEGIN BULK
$PARAM   POST    0
$PARAM   AUTOSPC YES
$
$
GRID     1       0       9.      0.0     2.5     0
GRID     2       0       14.5714315.714292.3571430
GRID     3       0       20.1428631.428572.2142860
GRID     4       0       25.7142947.142862.0714290
GRID     5       0       31.2857162.857141.9285710
GRID     6       0       36.8571478.571431.7857140
GRID     7       0       42.4285794.285711.6428570
GRID     8       0       48.     110.    1.5     0
GRID     9       0       36.     0.0     2.5     0
GRID     10      0       39.5714315.714292.3571430
GRID     11      0       43.1428631.428572.2142860
GRID     12      0       46.7142947.142862.0714290
GRID     13      0       50.2857162.857141.9285710
GRID     14      0       53.8571478.571431.7857140
GRID     15      0       57.4285794.285711.6428570
GRID     16      0       61.     110.    1.5     0
GRID     17      0       9.      0.0     -2.5    0
GRID     18      0       14.5714315.71429-2.357140
GRID     19      0       20.1428631.42857-2.214290
GRID     20      0       25.7142947.14286-2.071430
GRID     21      0       31.2857162.85714-1.928570
GRID     22      0       36.8571478.57143-1.785710
GRID     23      0       42.4285794.28571-1.642860
GRID     24      0       48.     110.    -1.5    0
GRID     25      0       36.     0.0     -2.5    0
GRID     26      0       39.5714315.71429-2.357140
GRID     27      0       43.1428631.42857-2.214290
GRID     28      0       46.7142947.14286-2.071430
GRID     29      0       50.2857162.85714-1.928570
GRID     30      0       53.8571478.57143-1.785710
GRID     31      0       57.4285794.28571-1.642860
GRID     32      0       61.     110.    -1.5    0
$
CROD     36      2       17      1
CROD     37      2       18      2
CROD     38      2       19      3
CROD     39      2       20      4
CROD     40      2       21      5
CROD     41      2       22      6
CROD     42      2       23      7
CROD     43      2       24      8
CROD     44      2       25      9
```

```
CROD    45      2       26      10
CROD    46      2       27      11
CROD    47      2       28      12
CROD    48      2       29      13
CROD    49      2       30      14
CROD    50      2       31      15
CROD    51      2       32      16
$
CQUAD4  1       1       1       2       10      9
CQUAD4  2       1       2       3       11      10
CQUAD4  3       1       3       4       12      11
CQUAD4  4       1       4       5       13      12
CQUAD4  5       1       5       6       14      13
CQUAD4  6       1       6       7       15      14
CQUAD4  7       1       7       8       16      15
CQUAD4  8       1       17      18      26      25
CQUAD4  9       1       18      19      27      26
CQUAD4  10      1       19      20      28      27
CQUAD4  11      1       20      21      29      28
CQUAD4  12      1       21      22      30      29
CQUAD4  13      1       22      23      31      30
CQUAD4  14      1       23      24      32      31
$
CSHEAR  15      3       1       2       18      17
CSHEAR  16      3       2       3       19      18
CSHEAR  17      3       3       4       20      19
CSHEAR  18      3       4       5       21      20
CSHEAR  19      3       5       6       22      21
CSHEAR  20      3       6       7       23      22
CSHEAR  21      3       7       8       24      23
CSHEAR  22      3       9       10      26      25
CSHEAR  23      3       10      11      27      26
CSHEAR  24      3       11      12      28      27
CSHEAR  25      3       12      13      29      28
CSHEAR  26      3       13      14      30      29
CSHEAR  27      3       14      15      31      30
CSHEAR  28      3       15      16      32      31
CSHEAR  29      3       2       10      26      18
CSHEAR  30      3       3       11      27      19
CSHEAR  31      3       4       12      28      20
CSHEAR  32      3       5       13      29      21
CSHEAR  33      3       6       14      30      22
CSHEAR  34      3       7       15      31      23
CSHEAR  35      3       8       16      32      24
$
$ THIS SECTION CONTAINS THE LOADS, CONSTRAINTS, AND CONTROL BULK DATA ENTRIES
$
$
SPC     1       2       456     0.0
SPC     1       3       456     0.0
SPC     1       4       456     0.0
SPC     1       5       456     0.0
SPC     1       6       456     0.0
SPC     1       7       456     0.0
SPC     1       8       456     0.0
SPC     1       10      456     0.0
SPC     1       11      456     0.0
SPC     1       12      456     0.0
SPC     1       13      456     0.0
```

```
SPC     1       14      456     0.0
SPC     1       15      456     0.0
SPC     1       16      456     0.0
SPC     1       18      456     0.0
SPC     1       19      456     0.0
SPC     1       20      456     0.0
SPC     1       21      456     0.0
SPC     1       22      456     0.0
SPC     1       23      456     0.0
SPC     1       24      456     0.0
SPC     1       26      456     0.0
SPC     1       27      456     0.0
SPC     1       28      456     0.0
SPC     1       29      456     0.0
SPC     1       30      456     0.0
SPC     1       31      456     0.0
SPC     1       32      456     0.0
SPC     1       1       123456  0.0
SPC     1       17      123456  0.0
SPC     1       9       123456  0.0
SPC     1       25      123456  0.0
$
$
$ THIS SECTION CONTAINS THE PROPERTY AND MATERIAL BULK DATA ENTRIES
$
$
PROD    2       1       .1
$
PSHEAR  3       1       .04
$
PSHELL  1       1       .05
$
MAT1    1       10.0+6          .333    2.591-4
$
$ Unit Force to check Statics
$
FORCE, 1, 8, , 0.25, 0.0, 0.0, 1.0
FORCE, 1,16, , 0.25, 0.0, 0.0, 1.0
FORCE, 1,24, , 0.25, 0.0, 0.0, 1.0
FORCE, 1,32, , 0.25, 0.0, 0.0, 1.0
$
EIGR    900     GIV     0.0     300.            3       0       1.-3    +EIGR
+EIGR   MAX
$
$   Z-DISPLACEMENTS ARE THE ONLY DEFORMATIONS NEEDED TO DEFINE A MODE SHAPE,
$   SO ALL OTHER DEGREES OF FREEDOM ARE LEFT OUT OF THE ANALYSIS SET.
$
ASET1   3       3       2       THRU    8
ASET1   3       3       10      THRU    16
$ Exclude grid points on the lower surface of the wing
$ASET1   3       3       18      THRU    24
$ASET1   3       3       26      THRU    32
$
ENDDATA
```

```
 ASSIGN DATABASE FREQ1 KIMBERLY NEW DELETE
EDIT NOLIST;
INSERT 2323
$ LOAD MODIFICATION $
PRINT("LOG=('         >>>DYNAMIC LOAD MODIFICATION')");
$ THE [PG] MATRIX REPLACES THE ORIGINAL GUST LOAD MATRIX [PDF]
  [PG] IS USED TO CALCULATE THE COMPLEX FREQUENCY RESPONSE MATRIX
  AT EACH REQUESTED FREQUENCY $
[PDF]:=[PG];
$ CALL UTMPRT(1,[PDF]);$
$ MODE SHAPES $
$ CALL UTMPRT(1,[PHIA]);$
INSERT 2327
$ MODAL COMPLEX FREQUENCY RESPONSE MATRIX: INCLUDES ALL FREQUENCIES $
CALL UTMPRT(1,[UFREQI]);
SOLUTION
  TITLE   =Beech King Air Horizontal Tail Model
  SUBTITLE=[Heta] Matrix (0 to 90 Hz): Mach=0.134, V=1860., QDP=25.3
  PRINT ROOT(MODES=ALL)=ALL
  ANALYZE
     BOUNDARY METHOD=5, REDUCE=3, SPC=1
     MODES
     FREQUENCY MODAL(DLOAD=10, FSTEP=20, DAMPING=6, GUST=60)
END
BEGIN BULK
$
$ THE STRUCTURAL MODEL
$
GRID    1       0       9.      0.0     2.5     0
GRID    2       0       14.5714315.714292.3571430
GRID    3       0       20.1428631.428572.2142860
GRID    4       0       25.7142947.142862.0714290
GRID    5       0       31.2857162.857141.9285710
GRID    6       0       36.8571478.571431.7857140
GRID    7       0       42.4285794.285711.6428570
GRID    8       0       48.     110.    1.5     0
GRID    9       0       36.     0.0     2.5     0
GRID    10      0       39.5714315.714292.3571430
GRID    11      0       43.1428631.428572.2142860
GRID    12      0       46.7142947.142862.0714290
GRID    13      0       50.2857162.857141.9285710
GRID    14      0       53.8571478.571431.7857140
GRID    15      0       57.4285794.285711.6428570
GRID    16      0       61.     110.    1.5     0
GRID    17      0       9.      0.0     -2.5    0
GRID    18      0       14.5714315.71429-2.357140
GRID    19      0       20.1428631.42857-2.214290
GRID    20      0       25.7142947.14286-2.071430
GRID    21      0       31.2857162.85714-1.928570
GRID    22      0       36.8571478.57143-1.785710
GRID    23      0       42.4285794.28571-1.642860
GRID    24      0       48.     110.    -1.5    0
GRID    25      0       36.     0.0     -2.5    0
GRID    26      0       39.5714315.71429-2.357140
GRID    27      0       43.1428631.42857-2.214290
GRID    28      0       46.7142947.14286-2.071430
GRID    29      0       50.2857162.85714-1.928570
GRID    30      0       53.8571478.57143-1.785710
```

```
GRID    31      0       57.4285794.28571-1.642860
GRID    32      0       61.     110.    -1.5    0
$
CROD    36      2       17      1
CROD    37      2       18      2
CROD    38      2       19      3
CROD    39      2       20      4
CROD    40      2       21      5
CROD    41      2       22      6
CROD    42      2       23      7
CROD    43      2       24      8
CROD    44      2       25      9
CROD    45      2       26      10
CROD    46      2       27      11
CROD    47      2       28      12
CROD    48      2       29      13
CROD    49      2       30      14
CROD    50      2       31      15
CROD    51      2       32      16
$
CQUAD4  1       1       1       2       10      9
CQUAD4  2       1       2       3       11      10
CQUAD4  3       1       3       4       12      11
CQUAD4  4       1       4       5       13      12
CQUAD4  5       1       5       6       14      13
CQUAD4  6       1       6       7       15      14
CQUAD4  7       1       7       8       16      15
CQUAD4  8       1       17      18      26      25
CQUAD4  9       1       18      19      27      26
CQUAD4  10      1       19      20      28      27
CQUAD4  11      1       20      21      29      28
CQUAD4  12      1       21      22      30      29
CQUAD4  13      1       22      23      31      30
CQUAD4  14      1       23      24      32      31
$
CSHEAR  15      3       1       2       18      17
CSHEAR  16      3       2       3       19      18
CSHEAR  17      3       3       4       20      19
CSHEAR  18      3       4       5       21      20
CSHEAR  19      3       5       6       22      21
CSHEAR  20      3       6       7       23      22
CSHEAR  21      3       7       8       24      23
CSHEAR  22      3       9       10      26      25
CSHEAR  23      3       10      11      27      26
CSHEAR  24      3       11      12      28      27
CSHEAR  25      3       12      13      29      28
CSHEAR  26      3       13      14      30      29
CSHEAR  27      3       14      15      31      30
CSHEAR  28      3       15      16      32      31
CSHEAR  29      3       2       10      26      18
CSHEAR  30      3       3       11      27      19
CSHEAR  31      3       4       12      28      20
CSHEAR  32      3       5       13      29      21
CSHEAR  33      3       6       14      30      22
CSHEAR  34      3       7       15      31      23
CSHEAR  35      3       8       16      32      24
$
$ THIS SECTION CONTAINS THE LOADS, CONSTRAINTS, AND CONTROL BULK DATA ENTRIES
```

```
$
$
SPC     1       2       456     0.0
SPC     1       3       456     0.0
SPC     1       4       456     0.0
SPC     1       5       456     0.0
SPC     1       6       456     0.0
SPC     1       7       456     0.0
SPC     1       8       456     0.0
SPC     1       10      456     0.0
SPC     1       11      456     0.0
SPC     1       12      456     0.0
SPC     1       13      456     0.0
SPC     1       14      456     0.0
SPC     1       15      456     0.0
SPC     1       16      456     0.0
SPC     1       18      456     0.0
SPC     1       19      456     0.0
SPC     1       20      456     0.0
SPC     1       21      456     0.0
SPC     1       22      456     0.0
SPC     1       23      456     0.0
SPC     1       24      456     0.0
SPC     1       26      456     0.0
SPC     1       27      456     0.0
SPC     1       28      456     0.0
SPC     1       29      456     0.0
SPC     1       30      456     0.0
SPC     1       31      456     0.0
SPC     1       32      456     0.0
SPC     1       1       123456  0.0
SPC     1       17      123456  0.0
SPC     1       9       123456  0.0
SPC     1       25      123456  0.0
$
$
$ THIS SECTION CONTAINS THE PROPERTY AND MATERIAL BULK DATA ENTRIES
$
$
PROD    2       1       .1
$
PSHEAR  3       1       .04
$
PSHELL  1       1       .05
$
MAT1    1       10.0+6          .333    2.591-4
$
$   Z-DISPLACEMENTS ARE THE ONLY DEFORMATIONS NEEDED TO DEFINE A MODE SHAPE,
$   SO ALL OTHER DEGREES OF FREEDOM ARE LEFT OUT OF THE ANALYSIS SET.
$
ASET1   3       3       2       THRU    8
ASET1   3       3       10      THRU    16
$ Exclude grid points on the lower surface of the wing
$ASET1   3       3       18      THRU    24
$ASET1   3       3       26      THRU    32
$
$ MODES AND FREQUENCIES
$
```

```
EIGR,5,INV,0.0,125.,3,3,,,EIGR
+IGR,MAX
$
$ FREQUENCY DEPENDENT LOADS GENERATION
$
DLOAD,10,1.0,1.0,30
RLOAD1,30,20,34
DLAGS,20,35
FORCE,35,8,,1.0,0.0,0.0,1.0
TABLED1,34,,,,,,,,TT1
+T1,0.0,1.0,1000.0,1.0
$ include frequency input data from freqinp1.dat
FREQ, 20,    .0000,    .0001,    .0002
FREQ, 20,   1.1822,   1.1823,   1.1824
FREQ, 20,   2.3645,   2.3646,   2.3647
FREQ, 20,   3.5467,   3.5468,   3.5469
FREQ, 20,   4.7290,   4.7291,   4.7292
FREQ, 20,   5.9113,   5.9114,   5.9115
FREQ, 20,   7.0936,   7.0937,   7.0938
FREQ, 20,   8.2759,   8.2760,   8.2761
FREQ, 20,   9.4581,   9.4582,   9.4583
FREQ, 20,  10.6404,  10.6405,  10.6406
FREQ, 20,  11.8227,  11.8228,  11.8229
FREQ, 20,  13.0050,  13.0051,  13.0052
FREQ, 20,  14.1873,  14.1874,  14.1875
FREQ, 20,  15.3695,  15.3696,  15.3697
FREQ, 20,  16.5518,  16.5519,  16.5520
FREQ, 20,  17.7341,  17.7342,  17.7343
FREQ, 20,  18.9164,  18.9165,  18.9166
FREQ, 20,  20.0987,  20.0988,  20.0989
FREQ, 20,  21.2809,  21.2810,  21.2811
FREQ, 20,  22.4632,  22.4633,  22.4634
FREQ, 20,  23.6455,  23.6456,  23.6457
FREQ, 20,  24.8278,  24.8279,  24.8280
FREQ, 20,  26.0101,  26.0102,  26.0103
FREQ, 20,  27.1923,  27.1924,  27.1925
FREQ, 20,  28.3746,  28.3747,  28.3748
FREQ, 20,  29.5569,  29.5570,  29.5571
FREQ, 20,  30.7392,  30.7393,  30.7394
FREQ, 20,  31.9215,  31.9216,  31.9217
FREQ, 20,  33.1037,  33.1038,  33.1039
FREQ, 20,  34.2860,  34.2861,  34.2862
FREQ, 20,  35.4683,  35.4684,  35.4685
FREQ, 20,  36.6506,  36.6507,  36.6508
FREQ, 20,  37.8329,  37.8330,  37.8331
FREQ, 20,  39.0151,  39.0152,  39.0153
FREQ, 20,  40.1974,  40.1975,  40.1976
FREQ, 20,  41.3797,  41.3798,  41.3799
FREQ, 20,  42.5620,  42.5621,  42.5622
FREQ, 20,  43.7443,  43.7444,  43.7445
FREQ, 20,  44.9265,  44.9266,  44.9267
FREQ, 20,  46.1088,  46.1089,  46.1090
FREQ, 20,  47.2911,  47.2912,  47.2913
FREQ, 20,  48.4734,  48.4735,  48.4736
FREQ, 20,  49.6557,  49.6558,  49.6559
FREQ, 20,  50.8379,  50.8380,  50.8381
FREQ, 20,  52.0202,  52.0203,  52.0204
FREQ, 20,  53.2025,  53.2026,  53.2027
FREQ, 20,  54.3848,  54.3849,  54.3850
```

```
FREQ, 20, 55.5671, 55.5672, 55.5673
FREQ, 20, 56.7493, 56.7494, 56.7495
FREQ, 20, 57.9316, 57.9317, 57.9318
FREQ, 20, 59.1139, 59.1140, 59.1141
FREQ, 20, 60.2962, 60.2963, 60.2964
FREQ, 20, 61.4785, 61.4786, 61.4787
FREQ, 20, 62.6607, 62.6608, 62.6609
FREQ, 20, 63.8430, 63.8431, 63.8432
FREQ, 20, 65.0253, 65.0254, 65.0255
FREQ, 20, 66.2076, 66.2077, 66.2078
FREQ, 20, 67.3899, 67.3900, 67.3901
FREQ, 20, 68.5721, 68.5722, 68.5723
FREQ, 20, 69.7544, 69.7545, 69.7546
FREQ, 20, 70.9367, 70.9368, 70.9369
FREQ, 20, 72.1190, 72.1191, 72.1192
FREQ, 20, 73.3013, 73.3014, 73.3015
FREQ, 20, 74.4835, 74.4836, 74.4837
FREQ, 20, 75.6658, 75.6659, 75.6660
FREQ, 20, 76.8481, 76.8482, 76.8483
FREQ, 20, 78.0304, 78.0305, 78.0306
FREQ, 20, 79.2127, 79.2128, 79.2129
FREQ, 20, 80.3949, 80.3950, 80.3951
FREQ, 20, 81.5772, 81.5773, 81.5774
FREQ, 20, 82.7595, 82.7596, 82.7597
FREQ, 20, 83.9418, 83.9419, 83.9420
FREQ, 20, 85.1241, 85.1242, 85.1243
FREQ, 20, 86.3063, 86.3064, 86.3065
FREQ, 20, 87.4886, 87.4887, 87.4888
FREQ, 20, 88.6709, 88.6710, 88.6711
FREQ, 20, 89.8532, 89.8533, 89.8534
VSDAMP,6,0.10
$
$ AERODYNAMIC MODEL
$
CAERO1,1,,,7,3,,,1,+AB
+AB,0.0,0.0,0.0,61.0,43.0,110.0,0.0,30.0
SPLINE1,3,,1,1,21,10
SET1,10,1,THRU,16
AERO,,45.5,1.0E-7
MKAERO1,1,0,0.134,,,,,,+CD
+CD,0.01,1.0,2.5,4.0,5.5,7.0,8.5,10.0
GUST,60,30,1.0E-4,0.0,1860.,25.3,0.134,,+GS1
+GS1,1,0
$
$ LOADING
$
$ THE [PG] MATRIX REPLACES THE ORIGINAL GUST LOAD MATRIX [PDF]
$ [PG] IS USED TO CALCULATE THE COMPLEX FREQUENCY RESPONSE MATRIX
$ AT EACH REQUESTED FREQUENCY
$ include load input data from dmiinp.dat
DMI,PG,CDP,REC,3,231,,,,DU0001
$ Group 1
+U0001,  1,1,1.0,0.0,  2,2,1.0,0.0,DU0002
+U0002,  3,3,1.0,0.0,  4,1,1.0,0.0,DU0003
+U0003,  5,2,1.0,0.0,  6,3,1.0,0.0,DU0004
+U0004,  7,1,1.0,0.0,  8,2,1.0,0.0,DU0005
+U0005,  9,3,1.0,0.0, 10,1,1.0,0.0,DU0006
+U0006, 11,2,1.0,0.0, 12,3,1.0,0.0,DU0007
+U0007, 13,1,1.0,0.0, 14,2,1.0,0.0,DU0008
```

```
+U0008, 15,3,1.0,0.0, 16,1,1.0,0.0,DU0009
+U0009, 17,2,1.0,0.0, 18,3,1.0,0.0,DU0010
+U0010, 19,1,1.0,0.0, 20,2,1.0,0.0,DU0011
+U0011, 21,3,1.0,0.0, 22,1,1.0,0.0,DU0012
+U0012, 23,2,1.0,0.0, 24,3,1.0,0.0,DU0013
+U0013, 25,1,1.0,0.0, 26,2,1.0,0.0,DU0014
+U0014, 27,3,1.0,0.0, 28,1,1.0,0.0,DU0015
+U0015, 29,2,1.0,0.0, 30,3,1.0,0.0,DU0201
$
$ Group 2
+U0201, 31,1,1.0,0.0, 32,2,1.0,0.0,DU0202
+U0202, 33,3,1.0,0.0, 34,1,1.0,0.0,DU0203
+U0203, 35,2,1.0,0.0, 36,3,1.0,0.0,DU0204
+U0204, 37,1,1.0,0.0, 38,2,1.0,0.0,DU0205
+U0205, 39,3,1.0,0.0, 40,1,1.0,0.0,DU0206
+U0206, 41,2,1.0,0.0, 42,3,1.0,0.0,DU0207
+U0207, 43,1,1.0,0.0, 44,2,1.0,0.0,DU0208
+U0208, 45,3,1.0,0.0, 46,1,1.0,0.0,DU0209
+U0209, 47,2,1.0,0.0, 48,3,1.0,0.0,DU0210
+U0210, 49,1,1.0,0.0, 50,2,1.0,0.0,DU0211
+U0211, 51,3,1.0,0.0, 52,1,1.0,0.0,DU0212
+U0212, 53,2,1.0,0.0, 54,3,1.0,0.0,DU0213
+U0213, 55,1,1.0,0.0, 56,2,1.0,0.0,DU0214
+U0214, 57,3,1.0,0.0, 58,1,1.0,0.0,DU0215
+U0215, 59,2,1.0,0.0, 60,3,1.0,0.0,DU0301
$
$ Group 3
+U0301, 61,1,1.0,0.0, 62,2,1.0,0.0,DU0302
+U0302, 63,3,1.0,0.0, 64,1,1.0,0.0,DU0303
+U0303, 65,2,1.0,0.0, 66,3,1.0,0.0,DU0304
+U0304, 67,1,1.0,0.0, 68,2,1.0,0.0,DU0305
+U0305, 69,3,1.0,0.0, 70,1,1.0,0.0,DU0306
+U0306, 71,2,1.0,0.0, 72,3,1.0,0.0,DU0307
+U0307, 73,1,1.0,0.0, 74,2,1.0,0.0,DU0308
+U0308, 75,3,1.0,0.0, 76,1,1.0,0.0,DU0309
+U0309, 77,2,1.0,0.0, 78,3,1.0,0.0,DU0310
+U0310, 79,1,1.0,0.0, 80,2,1.0,0.0,DU0311
+U0311, 81,3,1.0,0.0, 82,1,1.0,0.0,DU0312
+U0312, 83,2,1.0,0.0, 84,3,1.0,0.0,DU0313
+U0313, 85,1,1.0,0.0, 86,2,1.0,0.0,DU0314
+U0314, 87,3,1.0,0.0, 88,1,1.0,0.0,DU0315
+U0315, 89,2,1.0,0.0, 90,3,1.0,0.0,DU0401
$
$ Group 4
+U0401, 91,1,1.0,0.0, 92,2,1.0,0.0,DU0402
+U0402, 93,3,1.0,0.0, 94,1,1.0,0.0,DU0403
+U0403, 95,2,1.0,0.0, 96,3,1.0,0.0,DU0404
+U0404, 97,1,1.0,0.0, 98,2,1.0,0.0,DU0405
+U0405, 99,3,1.0,0.0,100,1,1.0,0.0,DU0406
+U0406,101,2,1.0,0.0,102,3,1.0,0.0,DU0407
+U0407,103,1,1.0,0.0,104,2,1.0,0.0,DU0408
+U0408,105,3,1.0,0.0,106,1,1.0,0.0,DU0409
+U0409,107,2,1.0,0.0,108,3,1.0,0.0,DU0410
+U0410,109,1,1.0,0.0,110,2,1.0,0.0,DU0411
+U0411,111,3,1.0,0.0,112,1,1.0,0.0,DU0412
+U0412,113,2,1.0,0.0,114,3,1.0,0.0,DU0413
+U0413,115,1,1.0,0.0,116,2,1.0,0.0,DU0414
+U0414,117,3,1.0,0.0,118,1,1.0,0.0,DU0415
+U0415,119,2,1.0,0.0,120,3,1.0,0.0,DU0501
```

```
$
$ Group 5
+U0501,121,1,1.0,0.0,122,2,1.0,0.0,DU0502
+U0502,123,3,1.0,0.0,124,1,1.0,0.0,DU0503
+U0503,125,2,1.0,0.0,126,3,1.0,0.0,DU0504
+U0504,127,1,1.0,0.0,128,2,1.0,0.0,DU0505
+U0505,129,3,1.0,0.0,130,1,1.0,0.0,DU0506
+U0506,131,2,1.0,0.0,132,3,1.0,0.0,DU0507
+U0507,133,1,1.0,0.0,134,2,1.0,0.0,DU0508
+U0508,135,3,1.0,0.0,136,1,1.0,0.0,DU0509
+U0509,137,2,1.0,0.0,138,3,1.0,0.0,DU0510
+U0510,139,1,1.0,0.0,140,2,1.0,0.0,DU0511
+U0511,141,3,1.0,0.0,142,1,1.0,0.0,DU0512
+U0512,143,2,1.0,0.0,144,3,1.0,0.0,DU0513
+U0513,145,1,1.0,0.0,146,2,1.0,0.0,DU0514
+U0514,147,3,1.0,0.0,148,1,1.0,0.0,DU0515
+U0515,149,2,1.0,0.0,150,3,1.0,0.0,DU0601
$
$ Group 6
+U0601,151,1,1.0,0.0,152,2,1.0,0.0,DU0602
+U0602,153,3,1.0,0.0,154,1,1.0,0.0,DU0603
+U0603,155,2,1.0,0.0,156,3,1.0,0.0,DU0604
+U0604,157,1,1.0,0.0,158,2,1.0,0.0,DU0605
+U0605,159,3,1.0,0.0,160,1,1.0,0.0,DU0606
+U0606,161,2,1.0,0.0,162,3,1.0,0.0,DU0607
+U0607,163,1,1.0,0.0,164,2,1.0,0.0,DU0608
+U0608,165,3,1.0,0.0,166,1,1.0,0.0,DU0609
+U0609,167,2,1.0,0.0,168,3,1.0,0.0,DU0610
+U0610,169,1,1.0,0.0,170,2,1.0,0.0,DU0611
+U0611,171,3,1.0,0.0,172,1,1.0,0.0,DU0612
+U0612,173,2,1.0,0.0,174,3,1.0,0.0,DU0613
+U0613,175,1,1.0,0.0,176,2,1.0,0.0,DU0614
+U0614,177,3,1.0,0.0,178,1,1.0,0.0,DU0615
+U0615,179,2,1.0,0.0,180,3,1.0,0.0,DU0701
$
$ Group 7
+U0701,181,1,1.0,0.0,182,2,1.0,0.0,DU0702
+U0702,183,3,1.0,0.0,184,1,1.0,0.0,DU0703
+U0703,185,2,1.0,0.0,186,3,1.0,0.0,DU0704
+U0704,187,1,1.0,0.0,188,2,1.0,0.0,DU0705
+U0705,189,3,1.0,0.0,190,1,1.0,0.0,DU0706
+U0706,191,2,1.0,0.0,192,3,1.0,0.0,DU0707
+U0707,193,1,1.0,0.0,194,2,1.0,0.0,DU0708
+U0708,195,3,1.0,0.0,196,1,1.0,0.0,DU0709
+U0709,197,2,1.0,0.0,198,3,1.0,0.0,DU0710
+U0710,199,1,1.0,0.0,200,2,1.0,0.0,DU0711
+U0711,201,3,1.0,0.0,202,1,1.0,0.0,DU0712
+U0712,203,2,1.0,0.0,204,3,1.0,0.0,DU0713
+U0713,205,1,1.0,0.0,206,2,1.0,0.0,DU0714
+U0714,207,3,1.0,0.0,208,1,1.0,0.0,DU0715
+U0715,209,2,1.0,0.0,210,3,1.0,0.0,DU0801
$
$ Group 8
+U0801,211,1,1.0,0.0,212,2,1.0,0.0,DU0802
+U0802,213,3,1.0,0.0,214,1,1.0,0.0,DU0803
+U0803,215,2,1.0,0.0,216,3,1.0,0.0,DU0804
+U0804,217,1,1.0,0.0,218,2,1.0,0.0,DU0805
+U0805,219,3,1.0,0.0,220,1,1.0,0.0,DU0806
+U0806,221,2,1.0,0.0,222,3,1.0,0.0,DU0807
```

```
+U0807,223,1,1.0,0.0,224,2,1.0,0.0,DU0808
+U0808,225,3,1.0,0.0,226,1,1.0,0.0,DU0809
+U0809,227,2,1.0,0.0,228,3,1.0,0.0,DU0810
+U0810,229,1,1.0,0.0,230,2,1.0,0.0,DU0811
+U0811,231,3,1.0,0.0
ENDDATA
```

```
 ASSIGN DATABASE FREQ1 KIMBERLY NEW DELETE
EDIT NOLIST;
INSERT 2323
$ MODE SHAPES $
CALL UTMPRT(1,[PHIA]);
INSERT 2327
PRINT("LOG=('          >>>DYNAMIC MODAL RESPONSE MODIFICATION')");
$ SET THE MODAL RESPONSES EQUAL TO UNITY TO COMPUTE THE STRESS
  AND DISP. DUE TO EACH MODE $
[UFREQI]:=[PG];
CALL UTMPRT(1,[UFREQI]);
SOLUTION
  TITLE   =Beech King Air Horizontal Tail Model
  SUBTITLE=Stresses and Disps. due to Each Mode
  PRINT ROOT(MODES=ALL)=ALL, DISP(RECT,FREQ=ALL)=7, STRE(RECT,FREQ=ALL)=17
  ANALYZE
     BOUNDARY METHOD=5, REDUCE=3, SPC=1
     MODES
     FREQUENCY MODAL(DLOAD=10, FSTEP=20, DAMPING=6, GUST=60)
END
BEGIN BULK
$
$ THE STRUCTURAL MODEL
$
GRID    1       0       9.      0.0     2.5     0
GRID    2       0       14.5714315.714292.3571430
GRID    3       0       20.1428631.428572.2142860
GRID    4       0       25.7142947.142862.0714290
GRID    5       0       31.2857162.857141.9285710
GRID    6       0       36.8571478.571431.7857140
GRID    7       0       42.4285794.285711.6428570
GRID    8       0       48.     110.    1.5     0
GRID    9       0       36.     0.0     2.5     0
GRID    10      0       39.5714315.714292.3571430
GRID    11      0       43.1428631.428572.2142860
GRID    12      0       46.7142947.142862.0714290
GRID    13      0       50.2857162.857141.9285710
GRID    14      0       53.8571478.571431.7857140
GRID    15      0       57.4285794.285711.6428570
GRID    16      0       61.     110.    1.5     0
GRID    17      0       9.      0.0     -2.5    0
GRID    18      0       14.5714315.71429-2.357140
GRID    19      0       20.1428631.42857-2.214290
GRID    20      0       25.7142947.14286-2.071430
GRID    21      0       31.2857162.85714-1.928570
GRID    22      0       36.8571478.57143-1.785710
GRID    23      0       42.4285794.28571-1.642860
GRID    24      0       48.     110.    -1.5    0
GRID    25      0       36.     0.0     -2.5    0
GRID    26      0       39.5714315.71429-2.357140
GRID    27      0       43.1428631.42857-2.214290
GRID    28      0       46.7142947.14286-2.071430
GRID    29      0       50.2857162.85714-1.928570
GRID    30      0       53.8571478.57143-1.785710
GRID    31      0       57.4285794.28571-1.642860
GRID    32      0       61.     110.    -1.5    0
$
CROD    36      2       17      1
```

```
CROD    37      2       18      2
CROD    38      2       19      3
CROD    39      2       20      4
CROD    40      2       21      5
CROD    41      2       22      6
CROD    42      2       23      7
CROD    43      2       24      8
CROD    44      2       25      9
CROD    45      2       26      10
CROD    46      2       27      11
CROD    47      2       28      12
CROD    48      2       29      13
CROD    49      2       30      14
CROD    50      2       31      15
CROD    51      2       32      16
$
CQUAD4  1       1       1       2       10      9
CQUAD4  2       1       2       3       11      10
CQUAD4  3       1       3       4       12      11
CQUAD4  4       1       4       5       13      12
CQUAD4  5       1       5       6       14      13
CQUAD4  6       1       6       7       15      14
CQUAD4  7       1       7       8       16      15
CQUAD4  8       1       17      18      26      25
CQUAD4  9       1       18      19      27      26
CQUAD4  10      1       19      20      28      27
CQUAD4  11      1       20      21      29      28
CQUAD4  12      1       21      22      30      29
CQUAD4  13      1       22      23      31      30
CQUAD4  14      1       23      24      32      31
$
CSHEAR  15      3       1       2       18      17
CSHEAR  16      3       2       3       19      18
CSHEAR  17      3       3       4       20      19
CSHEAR  18      3       4       5       21      20
CSHEAR  19      3       5       6       22      21
CSHEAR  20      3       6       7       23      22
CSHEAR  21      3       7       8       24      23
CSHEAR  22      3       9       10      26      25
CSHEAR  23      3       10      11      27      26
CSHEAR  24      3       11      12      28      27
CSHEAR  25      3       12      13      29      28
CSHEAR  26      3       13      14      30      29
CSHEAR  27      3       14      15      31      30
CSHEAR  28      3       15      16      32      31
CSHEAR  29      3       2       10      26      18
CSHEAR  30      3       3       11      27      19
CSHEAR  31      3       4       12      28      20
CSHEAR  32      3       5       13      29      21
CSHEAR  33      3       6       14      30      22
CSHEAR  34      3       7       15      31      23
CSHEAR  35      3       8       16      32      24
$
$ THIS SECTION CONTAINS THE LOADS, CONSTRAINTS, AND CONTROL BULK DATA ENTRIES
$
$
SPC     1       2       456     0.0
SPC     1       3       456     0.0
```

```
SPC      1       4       456     0.0
SPC      1       5       456     0.0
SPC      1       6       456     0.0
SPC      1       7       456     0.0
SPC      1       8       456     0.0
SPC      1       10      456     0.0
SPC      1       11      456     0.0
SPC      1       12      456     0.0
SPC      1       13      456     0.0
SPC      1       14      456     0.0
SPC      1       15      456     0.0
SPC      1       16      456     0.0
SPC      1       18      456     0.0
SPC      1       19      456     0.0
SPC      1       20      456     0.0
SPC      1       21      456     0.0
SPC      1       22      456     0.0
SPC      1       23      456     0.0
SPC      1       24      456     0.0
SPC      1       26      456     0.0
SPC      1       27      456     0.0
SPC      1       28      456     0.0
SPC      1       29      456     0.0
SPC      1       30      456     0.0
SPC      1       31      456     0.0
SPC      1       32      456     0.0
SPC      1       1       123456  0.0
SPC      1       17      123456  0.0
SPC      1       9       123456  0.0
SPC      1       25      123456  0.0
$
$
$ THIS SECTION CONTAINS THE PROPERTY AND MATERIAL BULK DATA ENTRIES
$
$
PROD     2       1       .1
$
PSHEAR   3       1       .04
$
PSHELL   1       1       .05
$
MAT1     1       10.0+6          .333    2.591-4
$
$   Z-DISPLACEMENTS ARE THE ONLY DEFORMATIONS NEEDED TO DEFINE A MODE SHAPE,
$   SO ALL OTHER DEGREES OF FREEDOM ARE LEFT OUT OF THE ANALYSIS SET.
$
ASET1    3       3       2       THRU    8
ASET1    3       3       10      THRU    16
$ Exclude grid points on the lower surface of the wing
$ASET1    3       3       18      THRU    24
$ASET1    3       3       26      THRU    32
$
$ MODES AND FREQUENCIES
$
GRIDLIST,7,8,16
ELEMLIST,17,QUAD4,1,8
EIGR,5,INV,0.0,125.,3,3,,,EIGR
+IGR,MAX
$
$ FREQUENCY DEPENDENT LOADS GENERATION
$
```

```
DLOAD,10,1.0,1.0,30
RLOAD1,30,20,34
DLAGS,20,35
FORCE,35,8,,1.0,0.0,0.0,1.0
TABLED1,34,,,,,,,,TT1
+T1,0.0,1.0,1000.0,1.0
FREQ, 20,  1., 2., 3.
VSDAMP,6,0.10
$
$ AERODYNAMIC MODEL
$
CAERO1,1,,,7,3,,,1,+AB
+AB,0.0,0.0,0.0,61.0,43.0,110.0,0.0,30.0
SPLINE1,3,,1,1,21,10
SET1,10,1,THRU,16
AERO,,45.5,1.0E-7
MKAERO1,1,0,0.134,,,,,,+CD
+CD,0.01,1.0,2.5,4.0,5.5,7.0,8.5,10.0
GUST,60,30,1.0E-4,0.0,1860.,25.3,0.134,,+GS1
+GS1,1,0
$
$ LOADING
$
$ THE [PG] MATRIX REPLACES THE ORIGINAL MODAL RESPONSE MATRIX [UFREQI] $
DMI,PG,CDP,REC,3,3,,,,DU0001
+U0001,  1,1,1.0,0.0,  2,2,1.0,0.0,DU0002
+U0002,  3,3,1.0,0.0
ENDDATA
```